Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi, Accredited by NAAC, with 'A' Grade)

Near Jnana Bharathi Campus, Bengaluru – 560056

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A Project Report on

"Realtime Weapon Surveillence using DeepLearning"

Submitted in partial fulfilment of the requirement for the award of the Degree of

Bachelor of Engineering
In
Computer Science and Engineering

Submitted by

SOURABH R	1DA17CS167
SPANDAN N	1DA17CS168
TANISHQ JAIN	1DA17CS180
TASMIYA KM	1DA17CS199

For the academic year 2020-21

Under the Guidance of

Dr. Nandini N

Associate Professor, Dept. of CSE, Dr.AIT, Bengaluru – 56.

Visvesvaraya Technological University

Jnana Sangama, Belagavi, Karnataka 590018.

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi, Accredited by NAAC, with 'A' Grade)

Near Jnana Bharathi Campus, Bengaluru – 560056

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the Project report entitled **"Real Time Weapon Surveillence using Deep Learning"** is carried out by **Sourabh R [1DA17CS167], Spandan Nepal [1DA17CS168], Tanishq Jain [1DA17CS180] and Tasmiya KM [1DA17CS199]**, bonafide students of **Dr. Ambedkar Institute of Technology, Bangalore-560056**, during the academic year 2020-2021 and is in partial fulfillment for the award of Degree in Bachelor of Engineering in Computer Science and Engineering Department. It is certified that all corrections/suggestions indicated during Internal Assessment have been incorporated in the report deposited in the department. The project report has been approved as it satisfies the academic requirements in respect of report work prescribed for the Bachelor of Engineering Degree.

-------------------------	-------------------------	-------------------------
Signature of the Guide	**Signature of the HOD**	**Signature of the Principal**
Dr. Nandini N,	**Dr. Siddaraju,**	**Dr. M Meenakshi,**
Associate Professor,	Professor and HEAD,	Principal,
Department of CSE,	Department of CSE,	Dr.AIT, Bengaluru – 56.
Dr.AIT, Bengaluru -56.	Dr.AIT, Bengaluru – 56.	

Viva-Voce Examination

Name of the Examiners	Signature with Date
1.	
2.	

ACKNOWLEDGEMENT

The satisfaction that accompanies to this project would be incomplete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made our efforts go in vain. We consider ourselves privileged to express our gratitude and respect towards all those who guided us through the project **“ REALTIME WEAPON SURVEILLENCE USING DEEPLEARNING ”.**

We would like to express our gratitude to **Dr. M. MEENAKSHI, Principal, Dr. AIT,** for providing the congenial environment to work in. We would like to express our gratitude to **Dr. SIDDARAJU, HOD, Dept. of Computer Science & Engineering, Dr. AIT,** for giving us the support, encouragement and providing us the required lab facilities that was necessary for the completion of this project.

As a token of gratitude, we would like to acknowledge our sincere gratefulness to the internal guide **Dr. Nandini N, Associate Professor, Department of CSE at Dr. AIT** for their unlimited support and encouragement provided throughout the process.

We also express our gratitude and sincere thanks to all the teaching and non-teaching staff of **Computer Science & Engineering Department.** Finally, yet importantly, we would like to express our heartfelt thanks to our beloved parents for their blessings and our friends for their help and wishes for the successful completion of this project report.

SOURABH R [1DA17CS167]
SPANDAN NEPAL [1DA17CS168]
TANISHQ JAIN [1DA17CS180]
TASMIYA KM [1DA17CS199]

ABSTRACT

Security and safety is a big concern for today’s modern world. For a country to be economically strong, it must ensure a safe and secure environment for investors and tourists. Having said that, Closed Circuit Television (CCTV) cameras are being used for surveillance and to monitor activities i.e. robberies but these cameras still require human supervision and intervention. We need a system that can automatically detect these illegal activities. Despite state-of-the-art deep learning algorithms, fast processing hardware, and advanced CCTV cameras, weapon detection in real-time is still a serious challenge.Observing angle differences, occlusions by the carrier of the firearm and persons around it further enhances the difficulty of the challenge.This work focuses on providing a secure place using CCTV footage as a source to detect harmful weapons by applying the state of the art open-source deep learning algorithms.Therefore, automatic gun detection is a prime requirement in current scenario and this paper presents automatic gun detection from cluttered scene using Convolutional Neural Networks (CNN). We have used Deep Convolutional Network (DCN), a state-of-the-art Faster Region-based CNN model, through transfer learning, for automatic gun detection from cluttered scenes.

We have implemented YOLO V3 “You Only Look Once” object detection model by training it on our customized dataset. The training results confirm that YOLO V3 outperforms YOLO V2 and traditional convolutional neural network (CNN). Applying this model in our surveillance system, we can attempt to save human life and accomplish reduction in the rate of manslaughter or mass killing. Additionally, our proposed system can also be implemented in high-end surveillance and security robots to detect a weapon or unsafe assets to avoid any kind of assault or risk to human life.

TABLE OF CONTENTS

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Violence committed with guns puts significant impact on public, health, psychological, and economic cost. Many people die each year from gun-related violence. Psychological trauma is frequent among children who are exposed to high levels of violence in their communities or through the media. India has more than 71 million firearms, the second highest in the world after the US. But only about 10 million of these are licensed and registered. Which means that despite having one of the strictest gun control laws in the world, around 86% of civilian firearms in the country are illegal. Unlike the US, India's law enforcement mechanism is weak. Today, most of the criminal activities are taken place using handheld arms particularly gun, pistol and revolver. Several surveys revealed that hand held gun is the foremost weapon used for diverse crimes like burglary, rape ,break-in, robbery, shoplifting , etc. The global death toll from use of guns may be as high as 1,000 dead each day. According to statistics, 4.2 in 100000 people are killed in India every year in mass shootings. Over the past eight years, the number of school mass shootings has increased drastically. Gun violence on school grounds resulted in many wounded or dead school kids which mirrors the gun violence problem in the United States. The failure to address the root cause of school gun violence is having lasting consequences for millions of American children. Applying this model in our surveillance system, we can attempt to save human life and accomplish reduction in the rate of manslaughter or mass killing. Additionally, our proposed system can also be implemented in high-end surveillance and security robots to detect a weapon or unsafe assets to avoid any kind of assault or risk to human life. Although the human visual framework is quick and precise and can likewise perform complex undertakings like distinguishing different items and recognizing snags with minimal cognizant idea, however, it is common truth that if an individual watches something very similar for quite a long time, there is an opportunity of sluggishness and lack of regard. Nowadays, with the accessibility of huge datasets, quicker GPU's, advanced machine learning algorithms, and better calculations, we can now effectively prepare PCs and develop automated computer-based system to distinguish and identify numerous items on a site with high accuracy. Recent developments indicate that machine learning and advance image processing algorithms have played dominant role in smart surveillances and security systems. Apart from this, popularity of smart devices and networked cameras has also empowered this domain.

1.1 Scope of The Work

Since decades, there are various approaches are heading towards the garbage management. Whereas our approach is unique and stable. As mentioned in the literature survey, many IEEE papers, journals have been produced, however they are individual implementations and didn't have an overall scope of the system. As shown in the Figure 1.2, there is integration of prediction of classes such as Handgun, Knife and Rifle along with time and Date adaptations are with the web and pyqt5 application. The figure also shows the photos of classes considered for the training of the model.

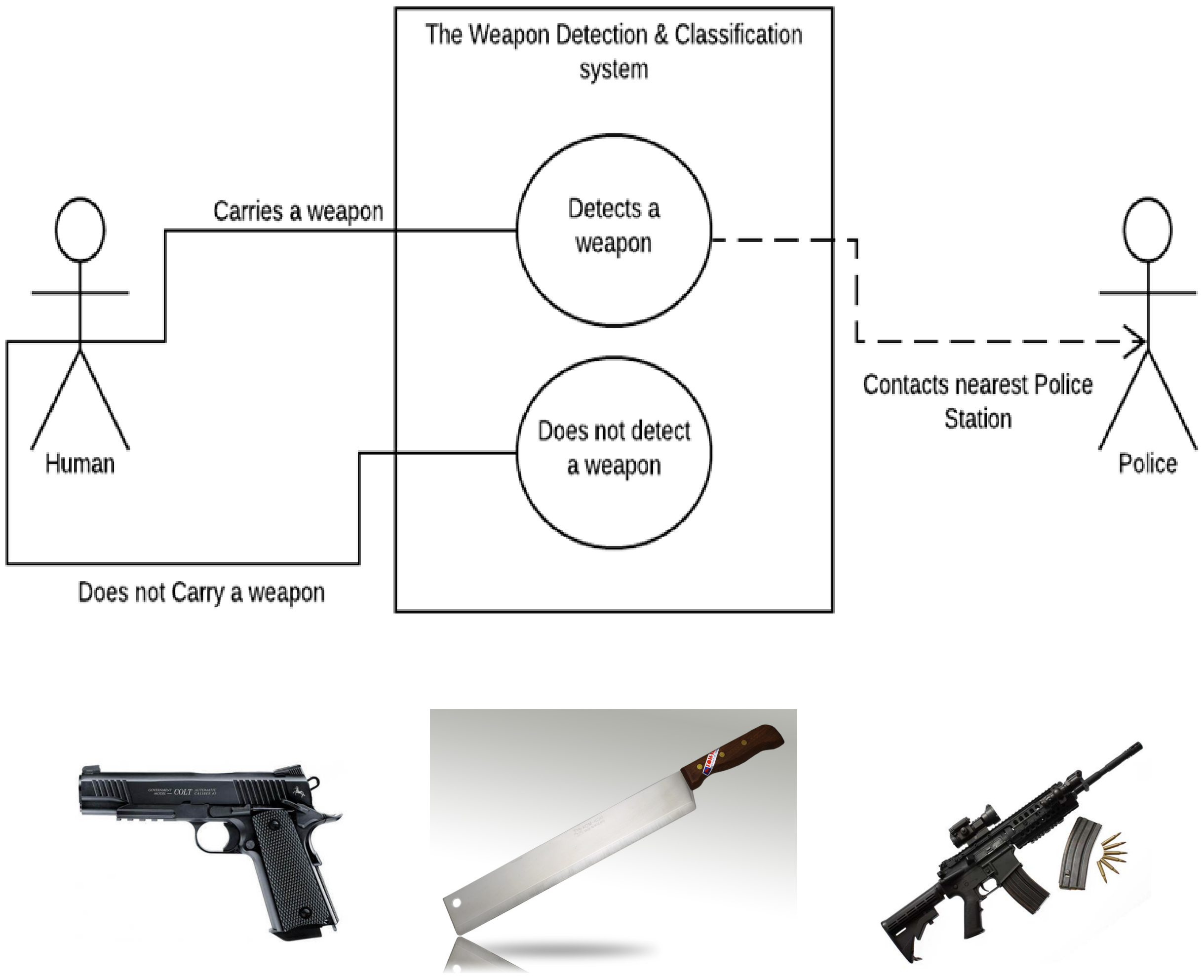


Figure 1.2 Scope of the work

1.2 Problem Definition

Today, most of the criminal activities are taken place using handheld arms particularly gun, pistol and revolver. The crime and social offence can be reduced by monitoring such activities. There is a need for surveillance system or control cameras to be deployed with automatic handheld gun detection and alert system.

1.3 Existing System

- There is no surveillance system developed to monitor cities
- Majority of the manual surveillance system still needs human eye to detect the abnormal activities and it takes a sufficient amount of time reporting to security officials to tackle the situation.
- Even after manual detection,there is no system developed to alert nearby police countys.

1.4 Objectives

The major objective of "Realtime weapon detection using deeplearning" is to provide a user-friendly, reliable and accessible system to reduce the crime by monitoring video feeds in real-time around the country. Finding a collection of images containing weapons belonging to Handgun,Rifle and Knife category.Since no standard Dataset is available for weapons, Building a Noval Dataset manually by combining Images from IMFDB and UGR Datset by adding Annotation to the Images.Building a Transfer Learning model by finding comparatively better Object Detection model .Training and Test the built model over custom dataset

1.5 Proposed System

- ➢ Develop a Client-Server Architecture based Product to implement our current Proposal
- ➢ Developing a web server for users to login and see detailed information like cctv, location,time,detected frame when the weapon is detected while in Surveillance
- ➢ Building a Client-GUI to feed and monitor video, as well to upload weapon detected frame
- ➢ Developing an ALERT System by emailing the client when weapon is detected

1.7 Summary

In this chapter, we discussed about the theme of the project and its current system. We also discussed about how to tackle the problems in current system. In the next chapter we will discuss about how we came to know about this problem by using certain surveys.

CHAPTER 2

LITERATURE SURVEY

For this project, we have referred following IEEE papers and website to analyze the requirements needed for the project and to know about the existing case diary. We are also analyzewhat are all the satisfying needs of a clients and to the public.

Primarily, research on gun detection focuses on Concealed Weapon Detection (CWD) and knife detection. CWD is stand on some techniques of imaging like infrared imaging, millimeter wave imaging, in application of luggage control at airports.The first and traditional sub-area in gun detection focuses on detecting concealed handguns in X-ray or millimetric wave images. The most representative application in this context is luggage control in airports. The existent methods achieve high accuracies by using different combinations of feature extractors and detectors, either using simple density descriptors [5], border detection and pattern matching [6] or using more complex methods such as cascade classifiers with boosting [13]. The effectiveness of these methods made them essential in some specific places. However, they have several limitations. As these systems are based on metal detection, they cannot detect non metallic guns. They are expensive to be used in many places as they require to be combined with X-ray scanners and Conveyor belts. They are not precise because they react to all metallic objects.

The second sub-area addresses gun detection in RGB images using classical methods. **The few existent papers essentially apply methods such as SIFT(Scale-Invariant Feature Transform) y RIFT (Rotation-Invariant Feature Transform), combined with Harris interest point detector or FREAK (Fast Retina Keypoint)**. For example, they developed an accurate software for pistol detection in RGB images. However, their method is unable to detect multiple pistols in the same scene. The used technique consists of rst, eliminating non related objects to a pistol from the segmented image using K-mean clustering algorithm then, applying SURF (Speeded Up Robust Features) method for detecting points of interest. Similarly, the authors in demonstrated that BoWSS (Bag of Words Surveillance System) algorithm has a high potential to detect guns. They rst extract features using SIFT, cluster the obtained functions using K-Means clustering and use SVM (Support Vector Machine) for the training. The authors in addresses rie detection in RGB images using SVM (Support Vector Machine).

All the above cited systems are slow,and not be used for constant monitoring, require the supervision of an operator and can not be used in open areas.

There is some existing research on weapon detection in image and video data as well. We have reviewed image classification research including the analysis of infrared data for concealed weapons [2], the detection of violent scenes in movie data [4], and gun detection in body camera videos. **While most research in the field have not employed deep learning/neural net techniques, other work by Olmos, Tabik, and Herrera investigate automatic gun detection in surveillance videos, triggering an alarm if the gun is detected (Automatic Handgun Detection Alarm in Videos Using Deep Learning) [3].** The project implements an RCNN in order to detect the action sequence of a handgun being drawn. Their implementation confirms the findings of Moez Baccouche and his team, who found that video action sequences could be effectively trained with 3-D CNNs (spatio-temporal features), and then classified using R-CNNs.[1] Olmos, Tabik, and Herreras research specifies further areas for improvement, including video preprocessing (adjusting contrast and luminosity to improve results). Our work includes Mask- RCNN, which is no where used for weapon detection.

R. Chellapa et.al. discussed briefly ject tracking and etection n surveillance cameras . The authors had explained he tracking f an object using multiple surveillance cameras. Another author ddressed techniques for tecting objects that come into contact with another object nd are occluded. They lso wrote regarding the egmentation of mean fluctuations. They outlined ow mean segmentation of shifts can help detect objects. hey used Bayesian Kalman filter with a simplified aussian blend BKF-SGM) algorithm to track the etected object .S Marques proposed distinct techniques for aluating e efficiency o istinct lgorithms for object ecognition B. Triggs et.al. described istogram oriented gradient (HOG). HOG became a novel architecture or feature extraction used mostly in applications involved human detection .In 2005, the sliding windo echnique was proposed for the recognition of number plates. They had used a sliding window for the urpose of egmentation and a eural network for haracter recognition on the number plate

With the development in CCTV's, object detection for different computer vision problems for real-time were performed and the idea to detect firearms were introduced first by L. Ward et al. in 2007 [30] and a surveillance system was also implemented by them a year later in 2008 [31].

In the aforementioned work, writers created an accurate pistol detection model for RGB pictures. However, in the same scene, their method did not detect various pistols [31-33]. The approach used comprises of first removing non-related items from the segmented picture using the K-mean clustering algorithm and then applying the SURF (Speed up Robust Features) method to detect points of interest. Darker gave the concept of SIFT based weapon detection algorithm and for ROI estimation, used the motion segmentation method [34]. SIFT algorithm is prone to false alarms, so for estimating ROI, authors used motion segmentation rather than using SIFT on complete image. When ROI was determined, then SIFT was applied to detect firearms in their case.

Different approaches then used for **weapon detection using sliding window and region proposal algorithms. HOG (Histogram of oriented Gradient) models were used to predict the objects in the frame. HOG significant work used low-level features, discriminative learning, and pictorial structure along with SVM** . These algorithms were slow for real-time scenarios with 14s per image. Although these classifiers gave good accuracies, the slowness of the sliding window method was a big problem, especially for the real-time implementation purpose

This work focuses on the state of the art deep learning network rather SIFT and HOG features which use handcrafted rules for feature extraction, selection, and detection in real-time visual scenario using CCTV cameras. X. Zhang et al. concluded an important finding that helped my work. They concluded that the automatic feature representation gave improved results rather than manual features [38]. Not only the learned features were better in performance, they also had learned the deep representation of the data and reduced a lot of manual work, and saved time and energy.

Rohith Vajhala et al. proposed the technique of knife and gun detection in surveillance systems. They had used HOG as a feature extractor along with backpropagation of artificial neural networks for classification purposes. The detection was performed using different scenarios, first weapon only and then using HOG and background subtraction methods for human before the desired object and claimed to have an accuracy of 83%.[39]. The aforementioned work uses the CNN along with non-linearity of ReLu, convolutional neural layer, fully connected layer, and dropout layer of CNN to reach a result for detection with multiple classes and implemented their work using the Tensor flow open-source platform. Their system achieved a test accuracy of 90.2 % for their dataset

Michał Grega et al. proposed knives and firearm detection in CCTV images. They had applied MPEG-7 and principle component analysis along with the sliding window approach, which made their work slower for real-time scenarios, although they claimed to achieve good accuracy on their test dataset.

Verma et al. had also used the deep learning technique to detect weapons and used the Faster RCNN model. The work was performed on imfdb, which in my opinion is not suitable to train a model for real-time case. They claimed to have an accuracy of 93.1% on that dataset but in the case of weapon detection, only achieving higher accuracy is not enough, and precision and recall must the considered [42]. Siham Tabik et al. work was very much related to the real-time scenario.

They used Faster RCNN to detect weapons in real-time using sliding window and region proposal methods. Best results were obtained by using the region proposal technique. The sliding window was also very time-consuming and took 14 s/image, on the other hand, the region proposal method processed the image in 140ms with 7 fps [43].

They trained the network on Faster RCNN using only one class focusing on reducing the false positive. Recent past objection detection work with the application to firearms was proposed in 2019, where a group of researchers, Javed Iqbal et alproposed orientation aware detection of the object. This system is more suitable for long and thin objects like rifles etc.

The predicted bounding box in their case was aligned with the object and had the less unnecessary area to deal with. Images of very high quality were used for training and testing purposes, which may make it less suitable for real-time scenarios [44]. Jose Luis Salazar Gonz´alez et al. work was very much related to achieve real-time results. They did immense experimentation using different datasets and trained Faster –RCNN using Feature Pyramid Network with Resnet50 and improves the previous state of the art by 3.91 %.

Pang et al. presented real-time concealed various object detection under human dress in [19]. Metallic guns on human skeleton were used for passive millimeter wave imagery which relies on YOLO algorithm on dataset of small scale. Subsequently, comparison is undertaken between Single MultiBox Detector algorithm, YOLOv3-13, SSD-VGG16, and YOLOv3-53 on PMMW dataset. Moreover, the weapon detection accuracy computed 36 frames per second of detection speed and 95% mean average precision. Warsi A et al. have contributed to automatically detecting the handgun in visual surveillance by implementing YOLO V3 algorithm with Faster Region-Based CNN (RCNN) by differentiating the number of false negatives and false positives [20], thus, taking real-time images and incorporating with ImageNet dataset then training it using YOLO V3 algorithm. They have compared Faster RCNN to YOLO V3 using four different videos and as a result YOLO V3 imparted faster speed in real-time environment.

Grega et al. presented an algorithm which automatically detects knives and firearms in CCTV image and alerts the security guard or operator [16]. **Majorly, focusing on limiting false alarms and providing a real-time application where specificity of the algorithm is 94.93% and sensitivity is 81.18% for knife detection.** Moreover, specificity for fire alarm system is 96.69% and sensitivity is 35.98% for different objects in the video. Mousavi et al. in [17] carried out video classifier also referred to as the Histogram of Directed Tracklets which identifies irregular conditions in complex scenes. In comparison to traditional approaches using optical flow which only measure edge features from two subsequent frames, descriptors have been developing over long-range motion projections called tracklets. Spatiotemporal cuboid footage sequences are statistically gathered on the tracklets that move through them.

Ji et al. developed a system for security footage which automatically identifies the human behavior using convolutional neural nets (CNNs) by forming deep learning model which operates directly on the raw inputs [18]. Therefore, 3D CNN model for classification requires the regularization of outputs with high-level characteristics to increase efficiency and integrating the observations of a variety of various models.

CHAPTER 3

SYSTEM SPECIFICATIONS AND REQUIREMENTS ANALYSIS

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of Testing's that describe user interactions that the software must provide. The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements, we need to have clear and thorough understanding of the products to be developed or being developed. An SRS minimize the time and effort required by developers to achieve desired goals and also minimizes the development cost.

3.1 Introduction to Software and Hardware components

3.1.1 Hardware Configuration

Hardware refers to the physical elements of the computer. This is sometimes called the machinery orequipment of the whole model.In contrast to software, hardware is a physical entity. Hardware and software are interconnected.Without software, the hardware of the system would have no function.Here is a list of hardware and software components used in our system along with their functionalitiesand use.

Hardware Requirements

- Processor: Minimum 1 GHz; Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi).
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above.

3.1.2 Software Configuration

In this chapter the software used and the language in which the program code is defined is mentioned and a program code dumping tools are explained. The chapter also documents the development of the program for the application. This program has been termed as "Source code". In this project, we have implemented coding using JavaScript and python language in Sublime Text editor. It's much easier working with this language. Because, it uses words that are easily recognized by humans. This typically results in a program that is much easier to write and understand.

Software Requirement

- Graphical Processing Unit
- OS: Windows 7 or higher
- Stack :Python 3.6,PyQt5
- Web Technology: HTML,CSS,JS
- Web Server: Django
- UGI for DB: SQLite

3.2 Purpose

The purpose of this SRS document is to provide a detailed overview of our software and hardware product, its parameters and goals. The document describes the projects target audience and its user interface, hardware and software requirements. Software requirements specification assures the project management stakeholders and client that the development team has really understood the business requirements documentation properly. This also provides confidence that the team will develop the functionality which has been detailed. The SRS is documented in such a way that it breaks the deliverables into smaller components. The information is organized in such a way that the developers will not only understand the boundaries within which they need to work, but also what functionality needs to be developed and in what order.

3.3 Feasibility Study

A project feasibility study is a comprehensive report that examines in detail the five frames of analysis of a given project. It also takes into consideration its four Ps, its risks and POVs, and its constraints (calendar, costs, and norms of quality). The goal is to determine whether the project should go ahead, be redesigned, or else abandoned altogether.The five frames of analysis are: The frame of definition; the frame of contextual risks; the frameof potentiality; the parametric frame; the frame of dominant and contingency strategies.

The four Ps are traditionally defined as Plan, Processes, People, and Power. The risks are considered to be external to the project (e.g., weather conditions) and are divided in eight categories:(Plan) financial and organizational (e.g., government structure for a private project); (Processes) environmental and technological; (People) marketing and socio-cultural; and (Power) legal and political.POVs are Points of Vulnerability: they differ from risks in the sense that they are internal to the projectand can be controlled or else eliminated.

The constraints are the standard constraints of calendar, costs and norms of quality that can each be objectively determined and measured along the entire project lifecycle. Depending on projects,portions of the study may suffice to produce a feasibility study; smaller projects, for example, may not require an exhaustive environmental assessment.

There are five areas of feasibility:

- Technical
- Economic
- Legal
- Operational
- Scheduling

3.3.1 Technical Feasibility

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project.

When writing a feasibility report, the following should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study.
- The part of the business being examined.
- The human and economic factor.
- The possible solutions to the problem.

The technical feasibility assessment is focused on gaining an understanding of the presenttechnical resources of the organization and their applicability to the expected needs of the proposedsystem. It is an evaluation of the hardware and software and how it meets the need of the proposedsystem.

3.3.2 Legal Feasibility

Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local data protection regulations and if the proposed venture is acceptable in accordance to the laws of the land.

3.3.3 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed developmentproject fits in with the existing business environment and objectives with regard to developmentschedule, delivery date, corporate culture and existing business processes.

3.3.4 Schedule Feasibility

A project will fail if it takes too long to be completed before it is useful. Typically, this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period.Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable.

3.3.5 Resource Feasibility

This involves questions such as how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies, and developmental procedures with company revenue prospectus.

3.3.6 Financial Feasibility

The financial viability of a project should provide the following information:

- Full details of the assets to be financed and how liquid those assets are.
- Rate of conversion to cash-liquidity (i.e., how easily the various assets can be converted to cash).
- Project&#39’s funding potential and repayment terms.

Sensitivity in the repayments capability to the following factors:

- Mild slowing of sales.
- Small increase in cost.
- Large increase in cost.
- Adverse economic conditions.

CHAPTER 4

SYSTEM DESIGN

The design of a project is where the key features of the project along with its structure are described. The main aim of our project is to provide a user-friendly android application and a helping hand to make our city clean.A responsible citizen can show his concern for keeping his surrounding clean with the help of our project.

4.1 UML Diagrams

UML stands for Unified Modeling Languages. UML is standardized general-purpose modeling language in the field of object oriented software engineering. The standard is managed, and was created by the Object Management group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprise of two major components: a Meta model and a notation. In future, some form of method or processed may also be added to are associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of the software projects.

GOALS:

The Primary in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extended the core concepts.
- Be independent of particular programming languages and development process.

- Provide a formal basis for understanding the modeling language.
- Encourage the growth of tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

4.1.1 Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in the term of actors, their goals (represented as use cases), and any dependencies between those use cases .

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

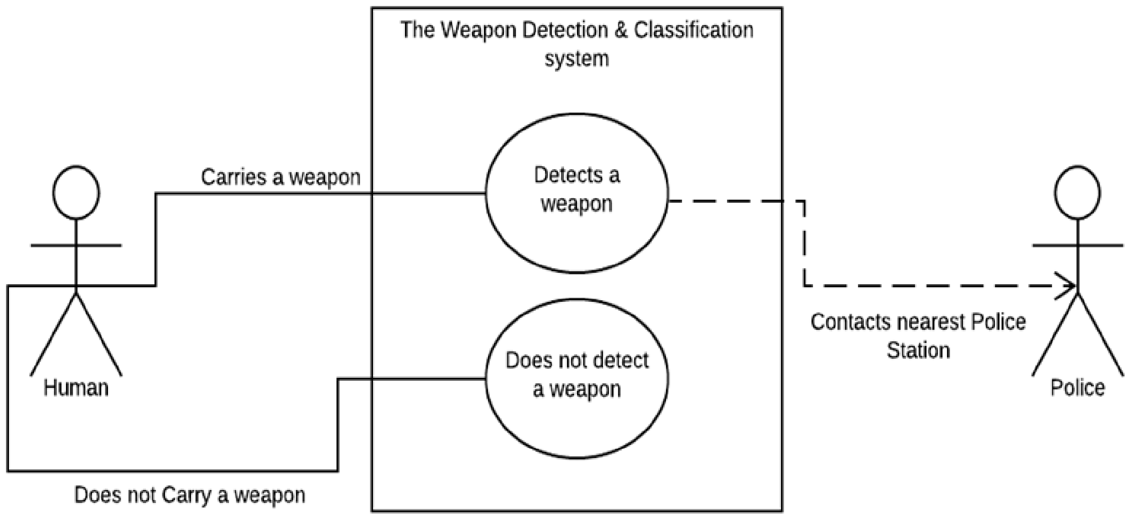


Figure 4.1 The Use Case Diagram

4.1.2 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

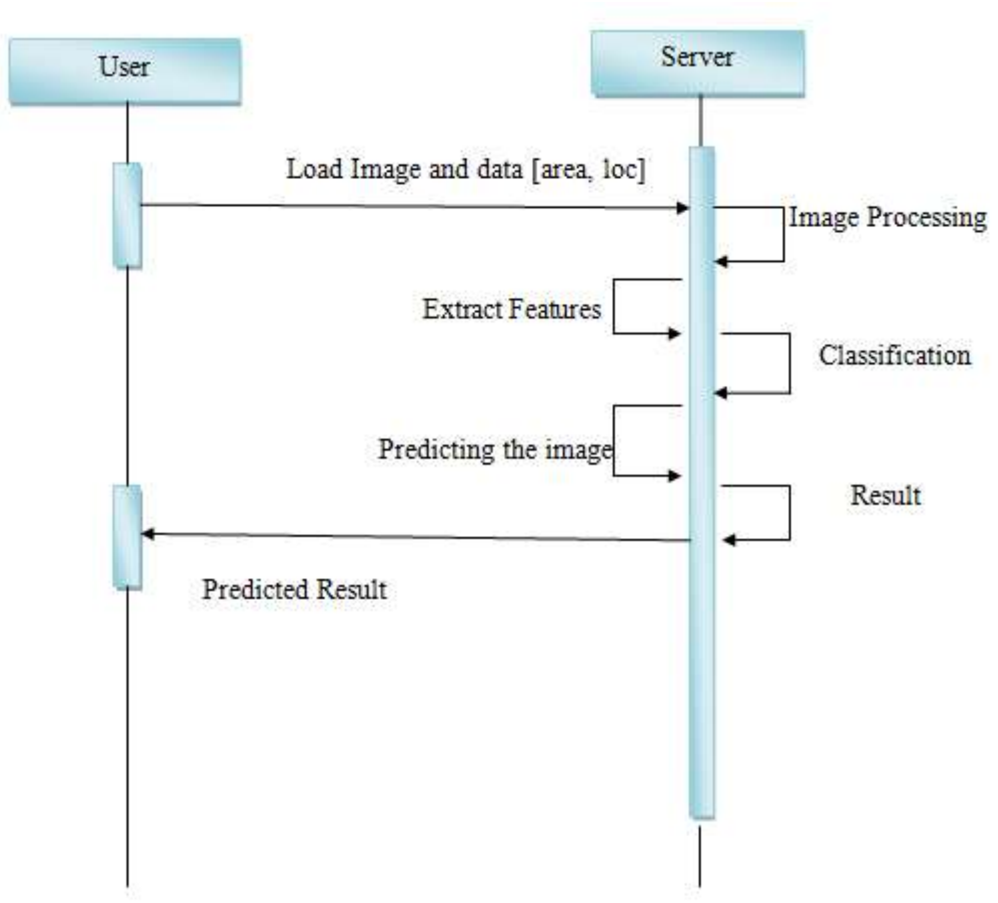
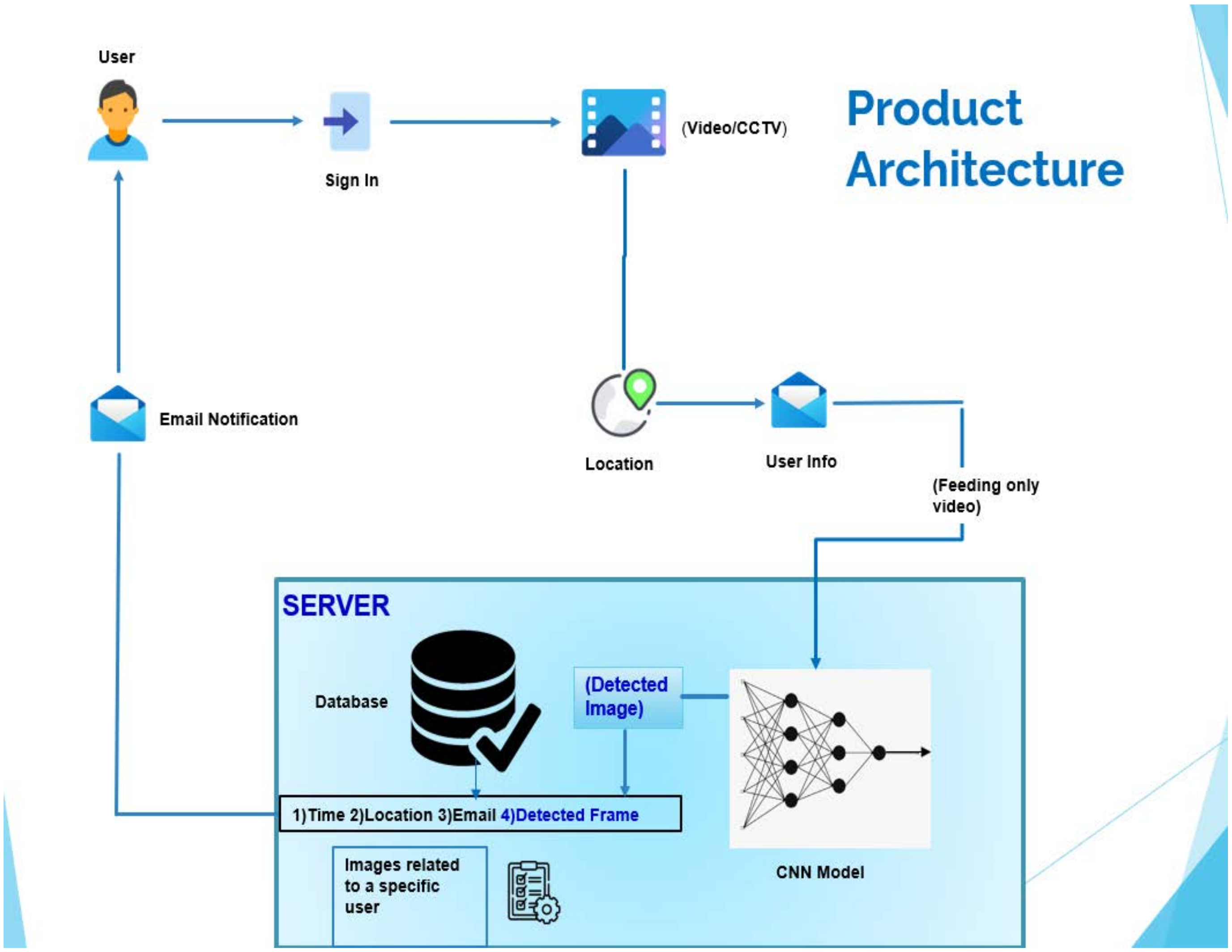
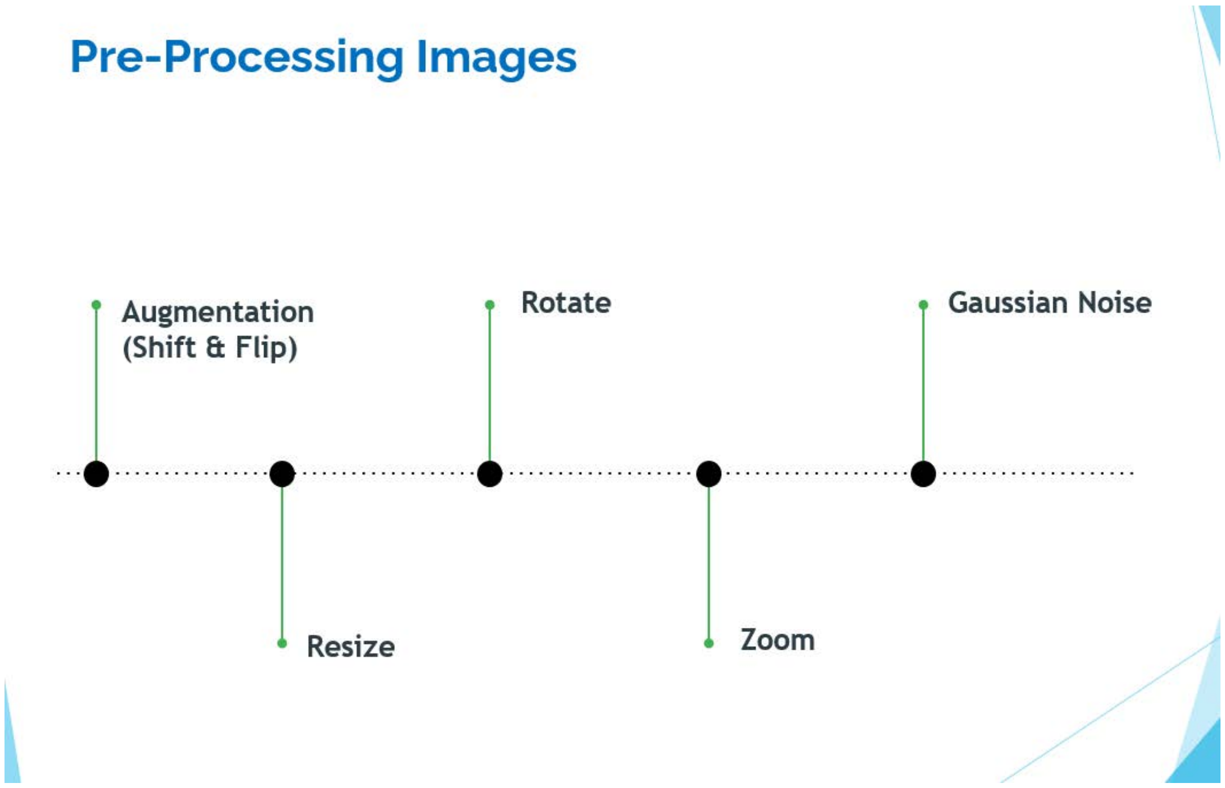


Figure 4.2 Sequence Diagram

4.2 Architecture and Design





Pre-Processing Images

Augmentation (Shift & Flip)

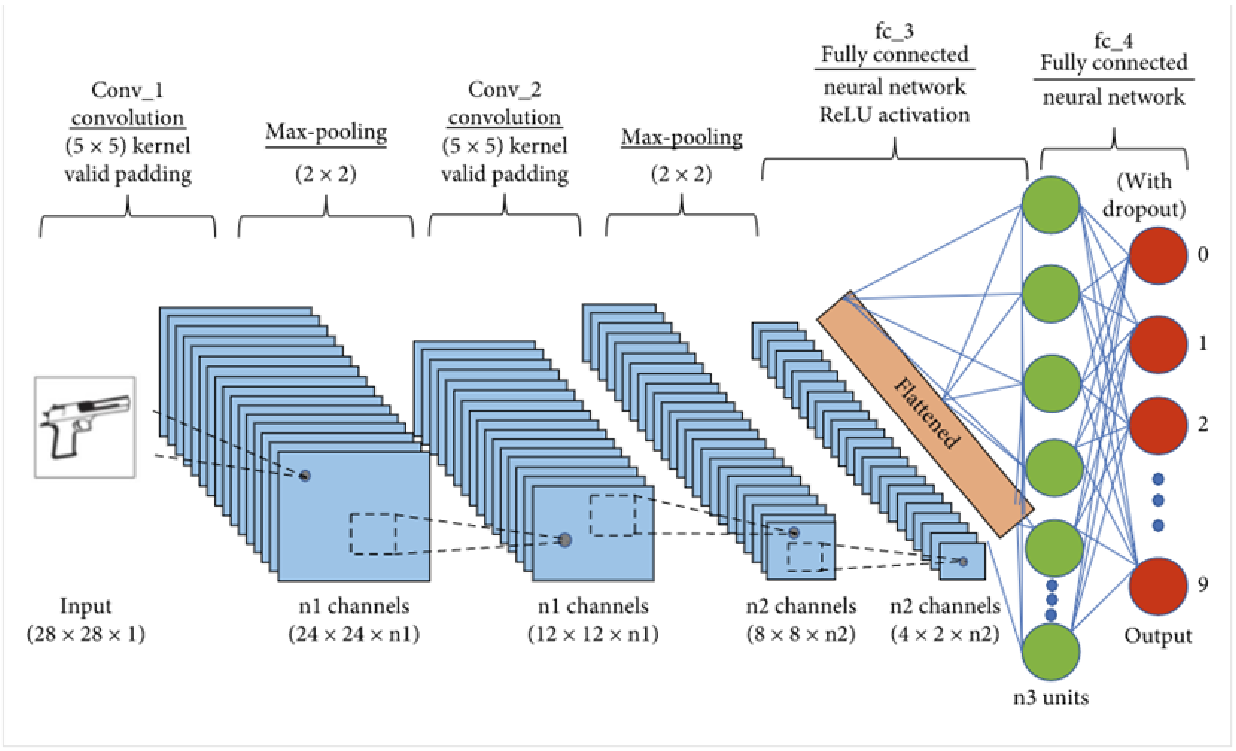
Resize

Rotate

Zoom

Gaussian Noise

CNN Architecture



4.3 Data Flow Diagram

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

Level: 0

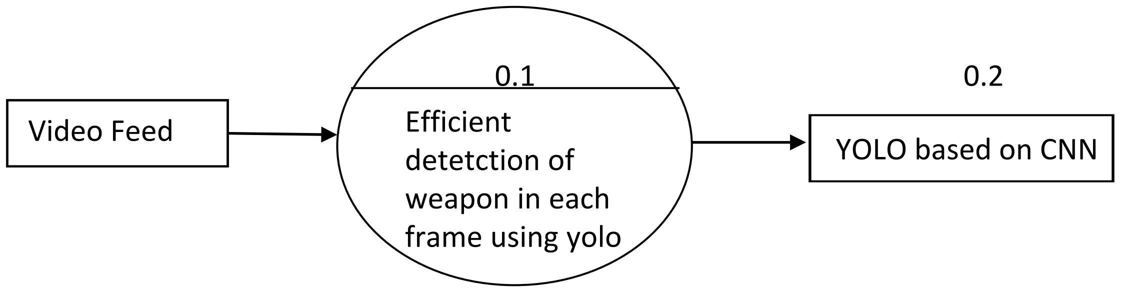


Figure 4.4 Level 0 Diagram

Level: 1

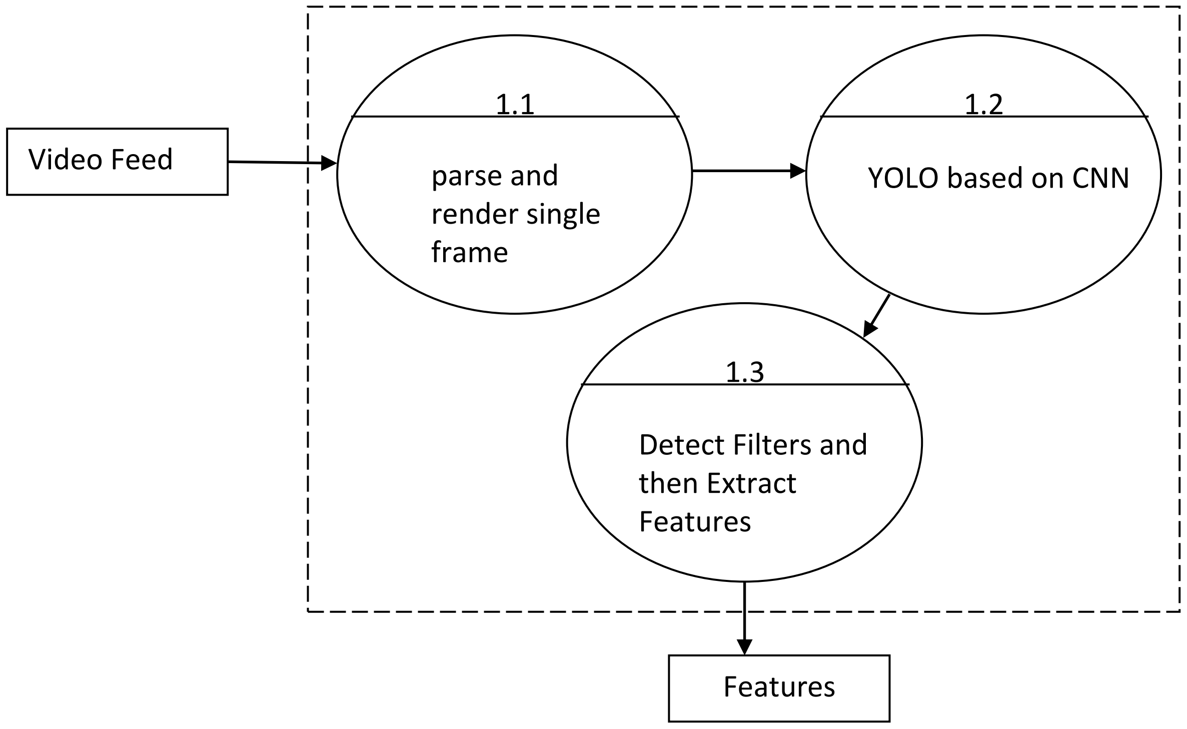

Video Feed
1.1
parse and render single frame
1.2
YOLO based on CNN
1.3
Detect Filters and then Extract Features
Features

Figure 4.5 Level 1 Diagram

4.4 Work Flow Diagram

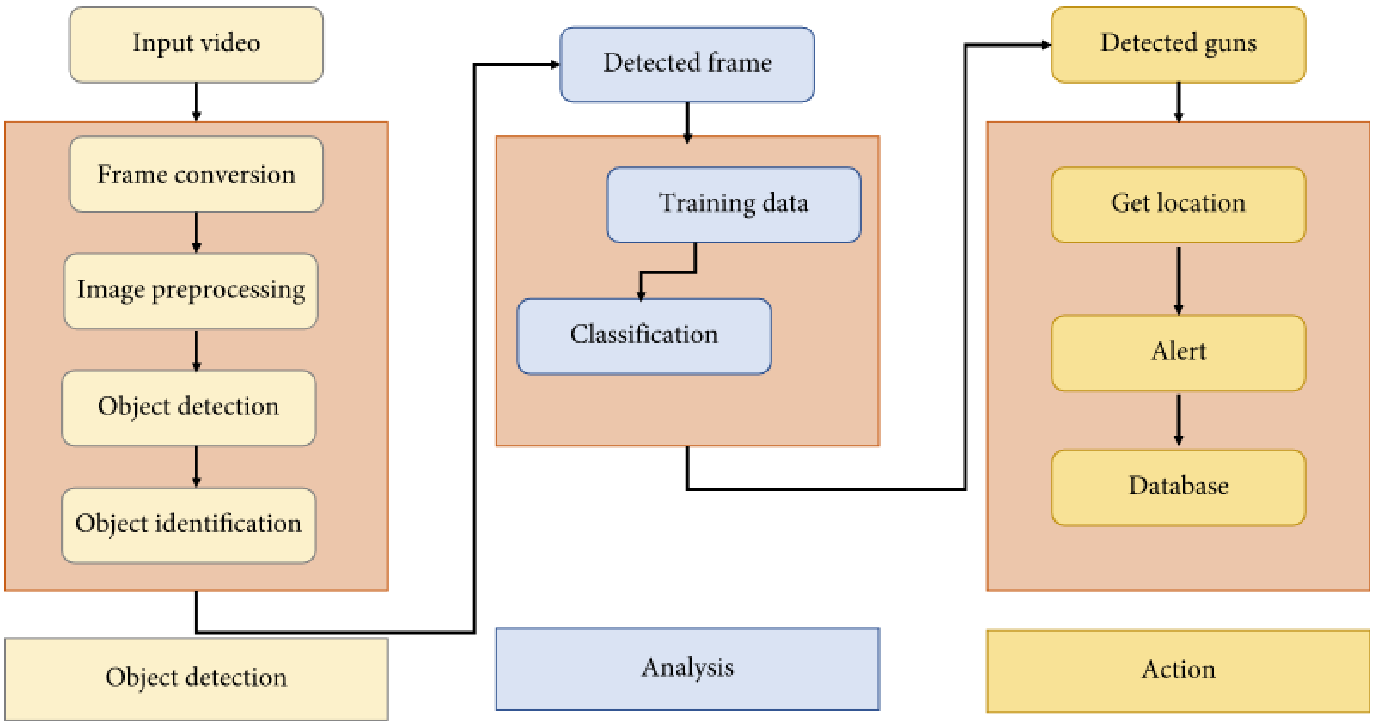


Figure 4.6: Work Flow Diagram

A work flow diagram is a basic visual layout of a business process. The above diagram shows the workflow of the our client-server architecture. The user who registers for the application is only allowed to continue further. Later after registering,he will login through registered account in client application . Once after login he will input details regarding video such as location of it being fed,the email id to which notification has to be sent and the video source. Currently we are using local video file, in future we could implement video streaming over web. As we move further to monitoring phase,this is how analysis works, video is converted to frame,and each frame is parsed through detection model. Before feeding frames, they undergo preprocessing involving adding/removing gaussian blur, reducing noise, rotating image to top-down position,if weapon is detected then alert will be sent over email. This could be improved by connecting with nearby security department.

4.5 Block Diagram for Object Detection

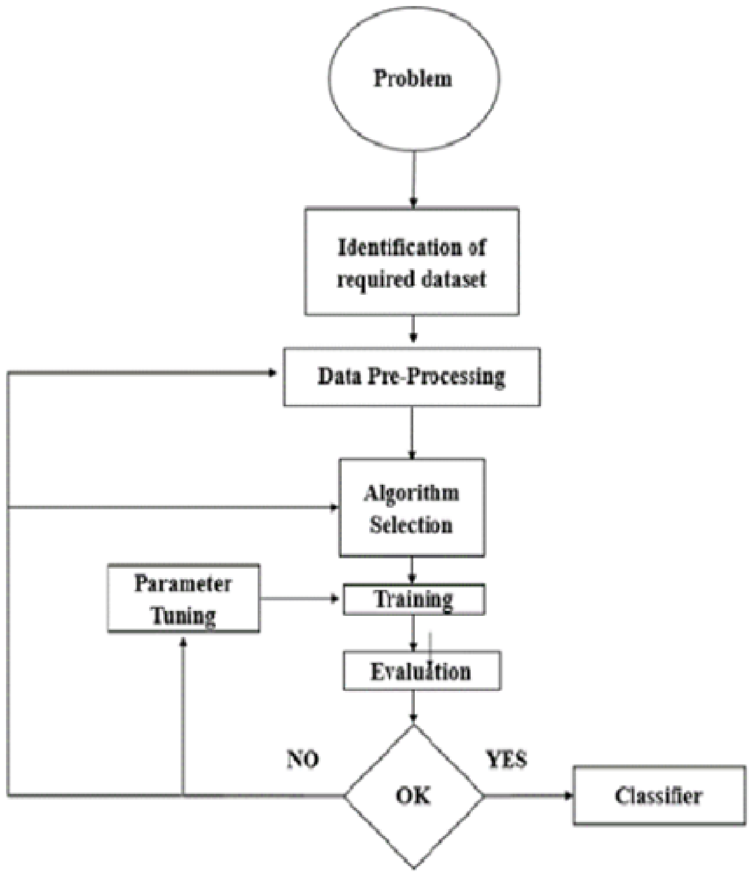


FIGURE 2. Training and Optimization Flow Diagram

The above block diagram describes the work flow of designing the prediction system being integrated with the python application. The blocks explain each process involved in the accomplishment of the system. At the beginning step, the Video feed will be given as an input to application.This feed could be in real -time from cctv or webcams.We trained the above model using YOLO object detection algorithm over custom dataset as well UGR Handgun dataset and IMFDB Dataset. Then the preprocessing was made on the set of images, before it was ready to be trained. Annotation, refers an explanation attached to objects of concern in an image. In our implementation, we manged to add annotation manually for all images. Along with that, an online annotation tool was used for the purpose. After annotation, the dataset is ready to be trained on the CNN Based YOLO Model. The is a deep neural network aimed to solve edge detection problem in machine learning or computer vision.

In other words, it can separate different objects in an image or a video. You give it an image, it gives you the object bounding boxes, classes and masks. The architecture of YOLO is provided in above diagram.As you can see a total of 53 layers are present commonly called as Darknet-53, backbone of YOLO Algorithm . CNN is a famous algorithm used for image classification using feature extraction and Classification using Dense Network

4.6 Backend Design

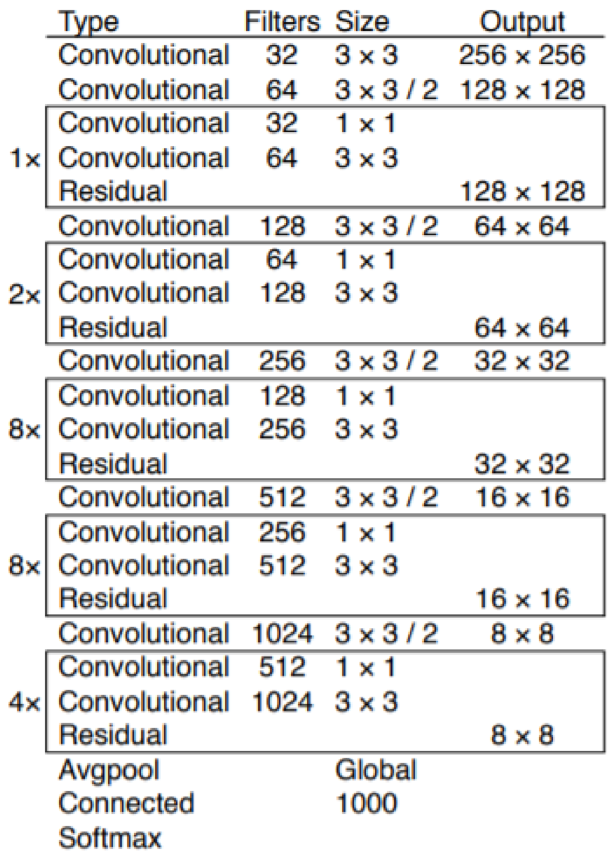
4.6.1 Python

The pride of conceiving Python goes to Guido van Rossum at Centrum Wiskunde&Informatica (CWI) in the late 1980s in the Netherlands capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.Python is popular for its simple syntax and short code length. Moreover, Python is also extremely versatile and well-designed. If that wasn’t enough, Python is a platform-independent language, meaning that software created using Python can be used on a wide variety of operating systems with no need of an interpreter. All of this means that programmers can spend a lot of the time they usually devote to getting the code to run or figuring out how it works tackling the more meaningful challenges of their particular development project.But Python doesn’t just facilitate the learning process, its readability also makes communication among programmers working on the same project smoother. This means that if a different programmer works on later additions to the code, they should have no problem understanding and working with the original code.Python has a vast set of libraries specific to Deep Learning, such as Keras, TensorFlow and Scikit-learn.

4.6.2 Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.
It supports SQLite database. Django offers dynamic HTML pages.Django is built for easy and simple projects and offers a Monolithic working style.The work flow would start from client end,where the client

will run the user app and then login through registered account. Then he specifies the details about video and which account to be notified.Later on Monitoring of video feed starts either directly from webcam or through some footage



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Table 1. **Darknet-53.**

Figure 4.8 showing the layers within Yolo Algorithm

4.7 Front-End Design.

4.7.1 HTML5

HTML is the language for publishing hypertext on the World Wide Web. This scripting language can be created and processed by a wide range of tools, from simple plain text editors - to sophisticated WYSIWYG (What You See Is What You Get) authoring tools. HTML uses tags to structure text into headings, paragraphs, lists, hypertext links etc.

A convenient way to automatically fix markup errors is to use the HTML Tidy utility. This makes the document easier to read and edit. Tidy is very effective at cleaning up markup created by authoring tools with sloppy habits. Tidy is able to fix up a wide range of problems and to bring to your attention things that you need to work on yourself. Each item found is listed with the line number and column so that you can see where the problem lies in your markup. Tidy won't generate a cleaned up version when there are problems that it can't be sure of how to handle. These are logged as "errors" rather than "warnings".

You specify which of these variants you are using by inserting a line at the beginning of the document. Each variant has its own DTD - Document Type Definition - which sets out the rules and regulations for using HTML. When you want to use a tool to validate the HTML document, the tool will know which variant you are using.

Transitional – Used when writing Web pages for the general public, when pages are accessible for viewing in older browsers. An advantage here is the support of style sheet, which include different features used in BODY such as bgcolor, text and link attributes.

4.7.2 JavaScript

JavaScript (JS) is a lightweight interpreted or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. Read more about JavaScript.

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about APIs specific to Web pages, please see Web APIs and DOM.

The standard for JavaScript is ECMAScript. As of 2012, all modern browsers fully support ECMAScript 5.1. Older browsers support at least ECMAScript 3. On June 17, 2015, ECMA International published the sixth major version of ECMAScript, which is officially called ECMAScript 2015, and was initially referred to as ECMAScript 6 or ES6. Since then, ECMAScript standards are on yearly release cycles. This documentation refers to the latest draft version, which is currently ECMAScript 2020.

4.7.3 CSS (**C**ascading **S**tyle **S**heets)

CSS is a language that describes the style of an HTML document.CSS describes how HTML elements should be displayed.It is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file, and reduce complexity and repetition in the structural content.

The advantages of using CSS are:

- Easier to maintain and update
- Greater consistency in design
- More formatting options

4.8 Summary

In this chapter, we discussed about various language adapted to design our website and data flow of our website through various dataflow diagrams, sequence diagrams and use case diagrams. In the next chapter, we will discuss about detailed implementation of our project.

CHAPTER 5

SYSTEM IMPLEMENTATION AND CODE

System Implementation uses the structure created during architectural design and the results of system analysis to construct system elements that meet the stakeholder requirements and system requirements developed in the early life cycle phases. These system elements are then integrated to form intermediate aggregates and finally the complete system of interest.

Implementation is the process that actually yields the lowest-level system elements in the system hierarchy (system breakdown structure). System elements are made, bought, or reused. Production involves the hardware fabrication processes of forming, removing, joining, and finishing, the software realization processes of coding and testing, or the operational procedures development processes for operators' roles. If implementation involves a production process, a manufacturing system which uses the established technical and management processes may be required.

The purpose of the implementation process is to design and create (or fabricate) a system element conforming to that element's design properties and/or requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition processes and the integration process.

A key difference between System Implementation and all other phases of the lifecycle is that all project activities up to this point have been performed in safe, protected, and secure environments, where project issues that arise have little or no impact on day-to-day business operations. Once the system goes live, however, this is no longer the case. Any miscues at this point will almost certainly translate into direct operational and/or financial impacts on the Performing Organization. It is through the careful planning, execution, and management of System Implementation activities that the Project Team can minimize the likelihood of these occurrences, and determine appropriate contingency plans in the event of a problem.

5.1 Code

5.1.1 Django Code(Server) :

Models (DAO)

```
# Changes uploaded file name
def Ascramble_uploaded_filename(instance, filename):
    """
    Scramble / uglify the filename of the uploaded file, but keep the files extension
(e.g., .jpg or .png)
    :param instance:
    :param filename:
    :return:
    """
    extension = filename.split(".")[-1]
    return "{}.{}".format(uuid.uuid4(), extension)



# Data model
class UploadAlert(models.Model):
    image = models.ImageField("Uploaded image", upload_to=scramble_uploaded_filename)
    user_ID = models.ForeignKey(Token, on_delete=models.CASCADE)
    alert_receiver = models.CharField(max_length=200)
    location = models.CharField(max_length=200)
    date_created = models.DateTimeField(auto_now_add=True)

# Generate and save a token each time a user is saved in a database
@receiver(post_save, sender=settings.AUTH_USER_MODEL)
def create_auth_token(sender,
                      instance=None,
                      created=False,
                      **kwargs):
    if created:
        Token.objects.create(user=instance)
```

Creating Forms

```
# User registration form
class CreateUserForm(UserCreationForm):

      email = forms.EmailField(required=True)

      class Meta:
            model = User
            fields = ['username', 'email', 'password1', 'password2']

      # Checks if the provided email exists
      def clean_email(self):
            if User.objects.filter(email=self.cleaned_data['email']).exists():
                  raise forms.ValidationError("The given email is already 
registered.")
            return self.cleaned_data['email']
```

Creating Filters

```
# Filtering
class DetectionFilter(django_filters.FilterSet):

      start_date = DateFilter(field_name="date_created", lookup_expr='gte')
      end_date = DateFilter(field_name="date_created", lookup_expr='lte')

      location = CharFilter(field_name='location', lookup_expr='icontains')
      alert_receiver = CharFilter(field_name='alert_receiver', 
lookup_expr='icontains')

      class Meta:
            model = UploadAlert
            fields = '__all__'
            exclude = ['customer', 'user_ID', 'image', 'uuid']
```

Creating API's

```
# Handles the registration page
def registerPage(request):

    if request.user.is_authenticated:
        return redirect('home')
    else:
        form = CreateUserForm()
        if request.method == 'POST':
            form = CreateUserForm(request.POST)
            if form.is_valid():
                form.save()
                user = form.cleaned_data.get('username')
                messages.success(request, 'Account was successfully created for ' +
user)
                return redirect('login')

        context = {'form':form}
        return render(request, 'detection/register.html', context)


# Handles the login page
def loginPage(request):
    if request.user.is_authenticated:
        return redirect('home')
    else:
        if request.method == 'POST':
            username = request.POST.get('username')
            password =request.POST.get('password')
            user = authenticate(request, username=username, password=password)
            if user is not None:
                login(request, user)
                return redirect('home')
            else:
                messages.info(request, 'Username OR password is incorrect')
        context = {}
        return render(request, 'detection/login.html', context)
```

```
# Handles the logout
def logoutUser(request):
    logout(request)
    return redirect('login')


# Handles the dashboard page
@login_required(login_url='login')
def home(request):

    token = Token.objects.get(user=request.user)

    uploadAlert = UploadAlert.objects.filter(user_ID = token)

    myFilter = DetectionFilter(request.GET, queryset=uploadAlert)
    uploadAlert = myFilter.qs

    context = {'myFilter':myFilter, 'uploadAlert':uploadAlert}

    return render(request, 'detection/dashboard.html', context)


# Handles the alert page
def alert(request,pk):

    uploadAlert = UploadAlert.objects.filter(image = str(pk) + ".jpg")

    myFilter = DetectionFilter(request.GET, queryset=uploadAlert)
    uploadAlert = myFilter.qs

    context = {'myFilter':myFilter, 'uploadAlert':uploadAlert}

    return render(request, 'detection/alert.html', context)
```

```
# Thread decorator definition
def start_new_thread(function):
    def decorator(*args, **kwargs):
        t = Thread(target = function, args=args, kwargs=kwargs)
        t.daemon = True
        t.start()
    return decorator

# Upload alert
@api_view(['POST'])
@permission_classes((IsAuthenticated, ))
def post_alert(request):
    serializer = UploadAlertSerializer(data=request.data)

    if serializer.is_valid():
        serializer.save()
        identify_email_sms(serializer)

    else:
        return "Error: Unable to process data!"

    return Response(request.META.get('HTTP_AUTHORIZATION'))

#Api to detect user provide email or phone number
def identify_email_sms(serializer):
    if(re.search('^[a-z0-9]+[\._]?[a-z0-9]+[@]\w+[.]\w{2,3}$',
serializer.data['alert_receiver'])):
        print("Valid Email")
        send_email(serializer)
    elif re.compile("[+3706][0-9]
{7}").match(serializer.data['alert_receiver']):
        # 1) Begins with +3706
        # 2) Then contains 7 digits
        print("Valid Mobile Number")
        send_sms(serializer)
    else:
        print("Invalid Email or Mobile number")
```

```
# Sends SMS
@start_new_thread
def send_sms(serializer):
    client = Client(settings.TWILIO_ACCOUNT_SID, settings.TWILIO_AUTH_TOKEN)

    message = client.messages.create(body=prepare_alert_message(serializer),
                                    from_=settings.TWILIO_NUMBER,
                                    to=serializer.data['alert_receiver'])

# Sends email
@start_new_thread
def send_email(serializer):
    send_mail('Weapon Detected!',
    prepare_alert_message(serializer),
    'xxxxx@gmail.com',
    [serializer.data['alert_receiver']],
    fail_silently=True,)

# Prepares the alert message
def prepare_alert_message(serializer):
    uuid_with_slashes = split(serializer.data['image'], ".")
    uuid = '.'.join(uuid_with_slashes)
    url = 'http://127.0.0.1:8000/alerts' + uuid
    print(url)
    return 'Weapon Detected! View alert at ' + url.split(".")[0]

# Splits string into a list
def split(value, key):
    return str(value).split(key)
```

Simple Serializer

```
# Serializer for UploadAlert Model
class UploadAlertSerializer(serializers.ModelSerializer):

    class Meta:
        model = UploadAlert
        fields = ('pk', 'image', 'user_ID', 'location', 'date_created',
'alert_receiver')
```

Url Paths/Routers

```
urlpatterns = [
    path('', views.home, name='home'),
    path('alerts/<str:pk>', views.alert, name='alert'),
    path('register/', views.registerPage, name='register'),
    path('login/', views.loginPage, name='login'),
    path('logout/', views.logoutUser, name='logout'),

    path('reset_password/',
        auth_views.PasswordResetView.as_view(template_name="detection/
password_reset.html"),
        name="reset_password"),

    path('reset_password_sent/',
        auth_views.PasswordResetDoneView.as_view(template_name="detection/
password_reset_sent.html"),
        name="password_reset_done"),

    path('reset/<uidb64>/<token>/',
        auth_views.PasswordResetConfirmView.as_view(template_name="detection/
password_reset_form.html"),
        name="password_reset_confirm"),

    path('reset_password_complete/',
        auth_views.PasswordResetCompleteView.as_view(template_name="detection/
password_reset_done.html"),
        name="password_reset_complete"),

   # Alert POST
    path('images/', views.post_alert, name='post_alert'),

    # Authentication
    url(r'^get_auth_token/$', rest_framework_views.obtain_auth_token,
name='get_auth_token'),
]
```

5.1.2 Client Application

Detection window

```
# Manages detection window, starts and stops detection thread
class DetectionWindow(QMainWindow):
      def __init__(self):
            super(DetectionWindow, self).__init__()
            loadUi('UI/detection_window.ui', self)

            self.stop_detection_button.clicked.connect(self.close)

      # Created detection instance
      def create_detection_instance(self, token, location, receiver
      ,file_p
      ):
            self.detection = Detection(token, location, receiver
            ,file_p
            )

      # Assigns detection output to the label in order to display detection output
      @pyqtSlot(QImage)
      def setImage(self, image):
            self.label_detection.setPixmap(QPixmap.fromImage(image))

      # Starts detection
      def start_detection(self):
            self.detection.changePixmap.connect(self.setImage)
            self.detection.start()
            self.show()

      # When closed
      def closeEvent(self, event):
            self.detection.running = False
            event.accept()
```

Login window

```
# LoginWindow class that manages login and opening the setting window
class LoginWindow(QMainWindow):
      def __init__(self):
            super(LoginWindow, self).__init__()
            loadUi('UI/login_window.ui', self)

            self.register_button.clicked.connect(self.go_to_register_page)
            self.login_button.clicked.connect(self.login)

            self.popup = QMessageBox()
            self.popup.setWindowTitle("Failed")

            self.show()

      # Open registration page
      def go_to_register_page(self):
            webbrowser.open('http://127.0.0.1:8000/register/')

      # Login function that manages the token authentication
      def login(self):
            try:
                  url = 'http://127.0.0.1:8000/api/get_auth_token/'
                  response = requests.post(url, data={'username':
self.username_input.text(),'password': self.password_input.text()})
                  json_response = json.loads(response.text)

                  # HTTP 200
                  if response.ok:
                        # Open settings window
                        self.open_settings_window(json_response['token'])
                  # Bad response
                  else:
                        # Show error
                        self.popup.setText("Username or Password is not correct")
                        self.popup.exec_()
```

```
except:
                  # Unable to access server
                  self.popup.setText("Unable to access server")
                  self.popup.exec_()

      # Opens settings window, passes the received token and closes login window
      def open_settings_window(self, token):
            self.settings_window = SettingsWindow(token)
            self.settings_window.displayInfo()
            self.close()
```

Setting window

```
# Manages the settings window
class SettingsWindow(QMainWindow):
      def __init__(self, token):
            super(SettingsWindow, self).__init__()
            loadUi('UI/settings_window.ui', self)

            self.token = token

            self.detection_window = DetectionWindow()

            self.pushButton.clicked.connect(self.go_to_detection)

            self.popup = QMessageBox()
            self.popup.setWindowTitle("Failed")
            self.popup.setText("Fields must not be empty.")


      def displayInfo(self):
            self.show()

      # Get input and go to detection window
      def go_to_detection(self):
            if self.location_input.text() == '' or self.sendTo_input.text() == '':
                  self.popup.exec_()
```

```
else:
                  if self.detection_window.isVisible():
                        print('Detection window is already open!')
                  else:
                        self.detection_window.create_detection_instance(self.token,
self.location_input.text(), self.sendTo_input.text()
                         ,self.send_file_name.text()
                         )
                        self.detection_window.start_detection()

      #When closed
      def closeEvent(self, event):
            if self.detection_window.isVisible():
                  self.detection_window.detection.running = False
                  self.detection_window.close()
                  event.accept()
```

Starting Application

```
# Starting the application
app = QApplication(sys.argv)
mainwindow = LoginWindow()

# Exiting
try:
      sys.exit(app.exec_())
except:
      print("Exiting")
```

5.1.3 Code to load config and weights and detect weapons

```
# Handles the YOLOv4 detection algorithm, saves detected frames and sends alert to
the server-side application
class Detection(QThread):


    def __init__(self, token, location, receiver,file_p):
        super(Detection, self).__init__()

        self.token = token
        self.location = location
        self.receiver = receiver
        self.file_p = file_p
        self.conif = 0.98
        self.x=0.8
        self.y=0.3


    changePixmap = pyqtSignal(QImage)


    # Runs the detection model, evaluates detections and draws boxes around
detected objects
    def run(self):


          # Loads Yolov4
        if("rifle" in self.file_p):
           print("video loaded")
           net = cv2.dnn.readNet("weights/yolo_rifle.weights", "cfg/
yolo_rifle.cfg")
           classes = ['Rifle']
           self.conif = 0.5
           self.x = 0.5
           self.y = 0.4
```

```
elif("knife" in self.file_p):
               print("video processing")
               net = cv2.dnn.readNet("weights/yolov3.weights", "cfg/yolo_v3.cfg")
               self.conif = 0.6
               self.x = 0.5
               self.y = 0.4
               with open('coco.names') as f:
                   classes =[line.strip() for line in f.readlines()]
        else:
           net = cv2.dnn.readNet("weights/yolov4.weights", "cfg/yolov4.cfg")
           with open("obj.names", "r") as f:
               classes = [line.strip() for line in f.readlines()]


        # Loads object names

        layer_names = net.getLayerNames()
        output_layers = [layer_names[i[0] - 1] for i in
net.getUnconnectedOutLayers()]
        colors = np.random.uniform(0, 255, size=(len(classes), 3))

        font = cv2.FONT_HERSHEY_PLAIN
        starting_time = time.time() - 11

        self.running = True

        # Starts camera
        cap = cv2.VideoCapture(self.file_p)
        cap.set(cv2.CAP_PROP_FPS, int(60))

        # Detection while loop
        while self.running:
            ret, frame = cap.read()
            if ret:

                height, width, channels = frame.shape
                # Running the detection model
```

```
  # Running the detection model
                blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0),
True, crop=False)
                net.setInput(blob)
                outs = net.forward(output_layers)

                # Evaluating detections
                class_ids = []
                confidences = []
                boxes = []
                for out in outs:
                    for detection in out:
                        scores = detection[5:]
                        class_id = np.argmax(scores)
                        confidence = scores[class_id]

                        # If detection confidance is above 98% a weapon was detected
                        if confidence > self.conif:

                            # Calculating coordinates
                            center_x = int(detection[0] * width)
                            center_y = int(detection[1] * height)
                            w = int(detection[2] * width)
                            h = int(detection[3] * height)
                            # Rectangle coordinates
                            x = int(center_x - w / 2)
                            y = int(center_y - h / 2)

                            boxes.append([x, y, w, h])
                            confidences.append(float(confidence))
                            class_ids.append(class_id)
                indexes = cv2.dnn.NMSBoxes(boxes, confidences, self.x, self.y)
                #Draw boxes around detected objects
                for i in range(len(boxes)):
                    if i in indexes:
                        x, y, w, h = boxes[i]
                        label = str(classes[class_ids[i]])
                        if(label!="knife" and "knife" in self.file_p):
                              continue
```

```
       confidence = confidences[i]
                        color = (256, 0, 0)
                        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
                        cv2.putText(frame, label + " {0:.1%}".format(confidence),
(x, y - 20), font, 3, color, 3)

                        elapsed_time = starting_time - time.time()

                        #Save detected frame every 10 seconds
                        if elapsed_time <= -10:
                            starting_time = time.time()
                            self.save_detection(frame)

                # Showing final result
                rgbImage = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                bytesPerLine = channels * width
                convertToQtFormat = QImage(rgbImage.data, width, height,
bytesPerLine, QImage.Format_RGB888)
                p = convertToQtFormat.scaled(854, 480, Qt.KeepAspectRatio)
                self.changePixmap.emit(p)

    # Saves detected frame as a .jpg within the saved_alert folder
    def save_detection(self, frame):
        cv2.imwrite("saved_frames/frame.jpg", frame)
        print('Frame Saved')
        self.post_detection()

    # Sends alert to the server
    def post_detection(self):

            url = 'http://127.0.0.1:8000/api/images/'
            headers = {'Authorization': 'Token ' + self.token}
            files = {'image': open('saved_frames/frame.jpg', 'rb')}
            data = {'user_ID': self.token,'location': self.location,
'alert_receiver': self.receiver}
            response = requests.post(url, files=files, headers=headers, data=data)
```

```
 # HTTP 200
           if response.ok:
               print('Alert was sent to the server')
           # Bad response
           else:
               print('Unable to send alert to the server')
```

Annotation of Dataset

```
<annotation>
  <folder>Handgun</folder>
  <filename>armas(3)</filename>
  <path>C:\Users\Sourabh\Desktop\Handguns\armas(3).jpg</path>
  <source>
      <database>Unknown</database>
  </source>
  <size>
      <width>1300</width>
      <height>866</height>
      <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
      <name>pistol</name>
      <pose>Unspecified</pose>
      <truncated>0</truncated>
      <difficult>0</difficult>
      <bndbox>
         <xmin>471</xmin>
         <ymin>207</ymin>
         <xmax>613</xmax>
         <ymax>359</ymax>
      </bndbox>
  </object>
</annotation>
```

CHAPTER 6

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. it is this process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not final in an unacceptable manner.

There are various types of test. Each test type addresses a specific testing requirement. An important points is that software testing should be distinguished from the separate discipline of software quality assurance, which encompasses all business process areas, not just testing. There are many approaches in software testing, but effective testing of complex products is essentially a process of investigation, not merely a matter of creating and following routine procedure.

One definition of testing is the process of questioning a product in order to evaluate it, where the questions are operations the tester attempts to execute with the product, and the product answers with its behavior in reaction to the probing of the tester. Although most of the intellectual processes of testing are nearly identical to that of review or inspection, the word testing is connected to mean the dynamic analysis of the product putting the product through its paces. Some of the common quality attribute include capability, reliability, efficiency, portability, maintainability, compatibility, compatibility and usability. A good test is sometimes described as one which reveals information of interest to someone who matters within the project community.

6.1 Testing Principles

In general, software engineers distinguish software faults from software failures. In case of a failures, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failures, a fault can also be described as an error in the correctness of the semantic of a computer program.

A fault will become a failure if exact computational conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software gets extended. Software testing is the technical investigation of the product under test to provide stakeholders with quality related information.

Software testing can be viewed as a sub-field of software quality assurance but optically exists independently. In SQA, software process specialists and auditors take a broader view on software and its development. They examine and change and change the software and its development.

They examine and change the software engineering process itself reduce the amount of faults that end up in the code or deliver faster.

A problem with software testing is that the number of defects in a software product can be very large, and the number of configuration of the product larger still, bugs that occur infrequently are difficult to find in testing. A rule of thumb is that a system that is expected to function without faults for a certain length of time must have already been tested for at least that length of time. This has severe consequences for projects to write long-lived reliable software.

A common practice of software testing is that it is performed by an independent group of testers after the functionality is developed but before it is shipped to the customer. This practice often results in the testing phase being used as project buffer to compensate for project delays. Another practice is to start software testing at the same moment the project starts and it is a continuous process until the project finishes.

Another common practice is for test suites to be developed during technical support escalation procedures, such tests are then maintained in regression testing suites to ensure that future updates to the software don't repeat any of the known mistakes, it is commonly believed that the earliest a defect is found the cheaper it is to fix it.

In counterpoint, some emerging software disciplines such as extreme programming and the agile software development movement. In this process, unit tests are written first, by the programmers. The test suites are continuously updated as new failure conditions and corner cases are discovered, and they are integrated with any regression tests that are developed. Unit tests are maintained along with the rest of the software source code and generally integrated into the build process. The software, tool, samples of data input and output, and configurations are all referred to collectively as a test harness.

6.2 Manual Testing

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing. Testers use test plans, test cases, or test scenarios to test software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

6.3 Automation Testing

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses software to test the product. This process involves automation of a manual process. Automation testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.

6.4 White box testing

White box testing is a testing technique that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

6.4.1 White Box Testing Techniques:

Statement Coverage - This technique is aimed at exercising all programming statements with minimal tests.

Branch Coverage - This technique is running a series of tests to ensure that all branches are tested at least once.

Path Coverage - This technique corresponds to testing all possible paths which means that each statement and branch is covered.

6.5 Black box testing

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. Independent Testing Team usually performs this type of testing during the software testing life cycle. This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.

There are different techniques involved in Black Box testing.

- Equivalence Class
- Boundary Value Analysis
- Domain Tests
- Orthogonal Arrays
- Decision Tables
- State Models
- Exploratory Testing
- All-pairs testing

6.6 PyQt5 App Testing

6.6.1 Functional Testing Test Cases

The functional testing of Mobiles normally consists in the areas of testing user interactions as well as testing the transactions:

- To validate whether all the required mandatory fields are working as required.
- To validate that the mandatory fields are displayed in the screen in a distinctive way than the non-mandatory fields.
- To validate whether the application works as per as requirement whenever the application starts/stops.

6.6.2 Usability Testing Test Cases

The usability testing process of the Mobile application is performed to have a quick and easy step application with less functionality than a slow and difficult application with many features. The main objective is to ensure that we end up having an easy-to-use, intuitive and similar to industry-accepted interfaces which are widely used.

- To ensure that the buttons should have the required size and be suitable to big fingers.

- To ensure that the buttons are placed in the same section of the screen to avoid confusion to the end users.
- To ensure that the icons are natural and consistent with the application.
- To ensure that the text is kept simple and clear to be visible to the users.
- To ensure that the short sentences and paragraphs are readable to the end users.
- To ensure that the font size is big enough to be readable and not too big or too small.

The purpose of this document is to provide overview of the testing, plus the techniques. Three levels of software testing is done at various software development life cycle phase.

- **Unit Testing**: In which each unit of the software is tested to verify that the detailed design for the unit has been correctly implemented.
- **Integration Testing**: In which progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a whole.
- **System Testing**: In which the software is integrated to the overall product and tested to show that all requirements are met. A further level of testing is also done in accordance with requirements.
- **Acceptance Testing**: Upon which the acceptance of the complete software is based. The clients often do this.
- **Usability Testing**: Is used to refer the repetition of the earlier successful tests to ensure that changes made in the software have not introduced new bugs.

In recent years the term grey box testing has come into common usage. The typical grey box tester is permitted to set up or manipulate the testing environment, like seeding a database, and can view the state of the product after his actions, like performing SQL query on the database to be certain of the values of columns.

6.7 Unit Testing

Unit testing is done at the development level. When a developer builds a piece of code that delivers a set of functionality, the developer tests it to make sure it works and delivers the required functionality. It is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a WhiteBox testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All buttons (features) must function properly.
- Pages must be activated from the identified link.
- The entry screen, video rendering and responses must not be delayed.

Features to be tested

- All buttons should take the user to the correct layout.
- All values should be stored into the database correctly.
- All links should take the user to the correct page.
- All the api's should provide Authentication error if accessed without Token

6.8 INTEGRATION TESTING

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. All the modules of the system are integrated together and an integration test plan is executed based on the sequence of activities defined previously. Integration test plan also checks if there is a smooth and proper communication between the modules of the system.

There are three types of integration testing:

- **Top-Down Integration**: Top down integration is an incremental approach to construction of program structures. Modules are integrated by moving downwards throw the control hierarchy beginning with the main control module.
- **Bottom-Up Integration**: Bottom up integration as in name implies, begins construction and testing with automatic modules.
- **Regression Testing**: In this content of an integration test strategy, regression testing is the re execution of some subset of test that have already been conducted to ensure that changes have not propagated unintended side effects.

6.9 SYSTEM TESTING

System testing includes a series of different tests that verify all system elements that have been properly integrated to perform allocated function. System elements shall include interfaces between the other system APIs, which would work together. The modules together constitute the overall system. All the functionalities of the proposed system are checked for the proper coordination and working as per the requirement specified. System testing means testing for the configuration of the system. Checking whether the system will works as we decided or not after the integration of the modules.

6.9.1 Acceptance testing

Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application. Here we will check for the acceptance of the inputs (image, location, area).

6.9.2 Usability testing User-friendliness check.

Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing. It includes detection of errors in the application. The testing process starts with a test plan that recognizes test-related activities, such as test case generation, testing criteria, and resource allocation for testing. The code is tested and mapped against the design document created in the design phase. The output of the testing phase is a test report containing errors that occurred while testing the application. Testing has been done for each of the individual activities of the project.

CHAPTER 7

RESULTS AND SNAPSHOTS

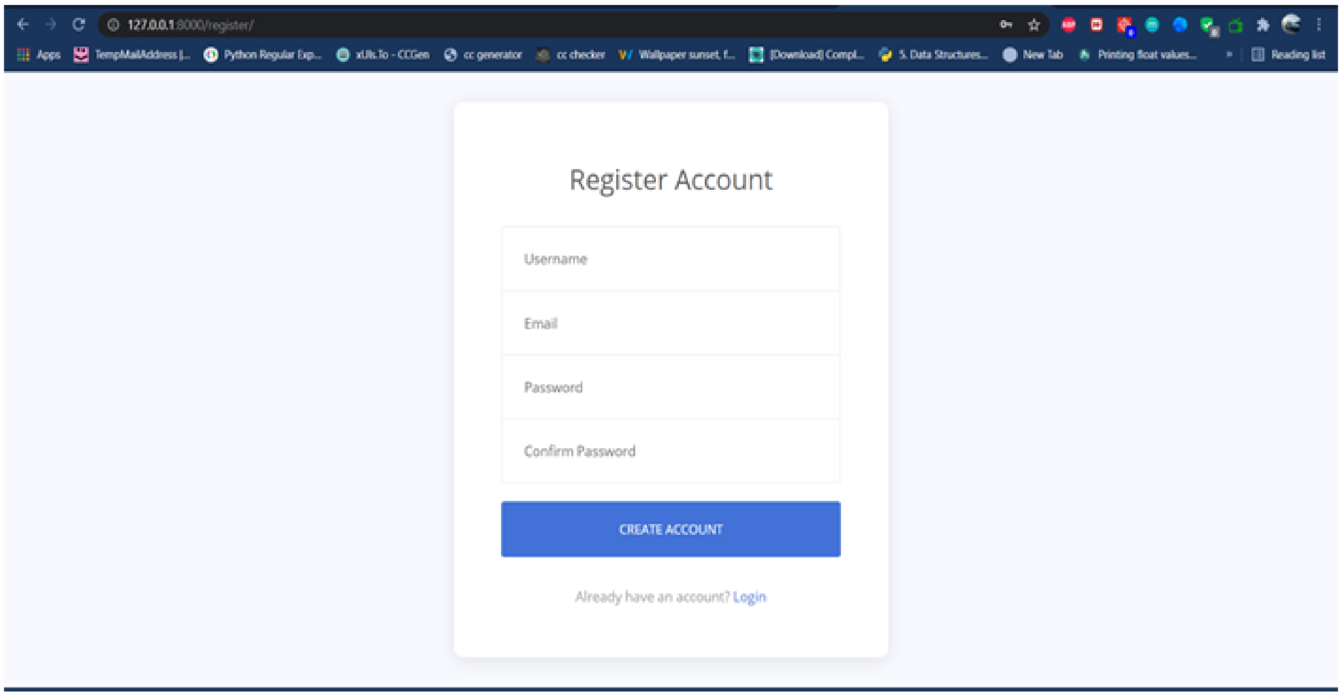
7.1 Server Application

- **Step: 1 Run the Django Server**

You can run server either by using ngx or wsgi

- **Step: 2 Register or Sign Up**

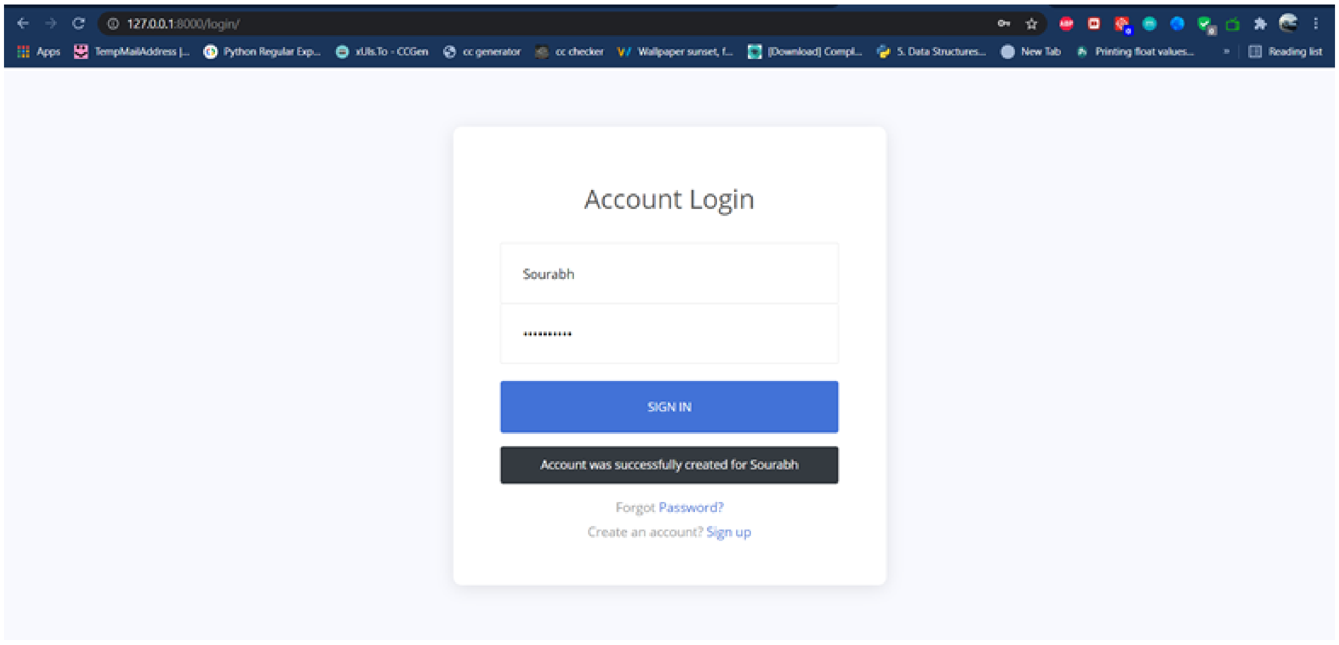
Create an Account by providing Email and Password.



➢ **Step: 3 Login**

Login using Email and Password.

➢ **Now run client Application and provide video feed source and information about the location and email.**



Account Login

Sourabh

••••••••••

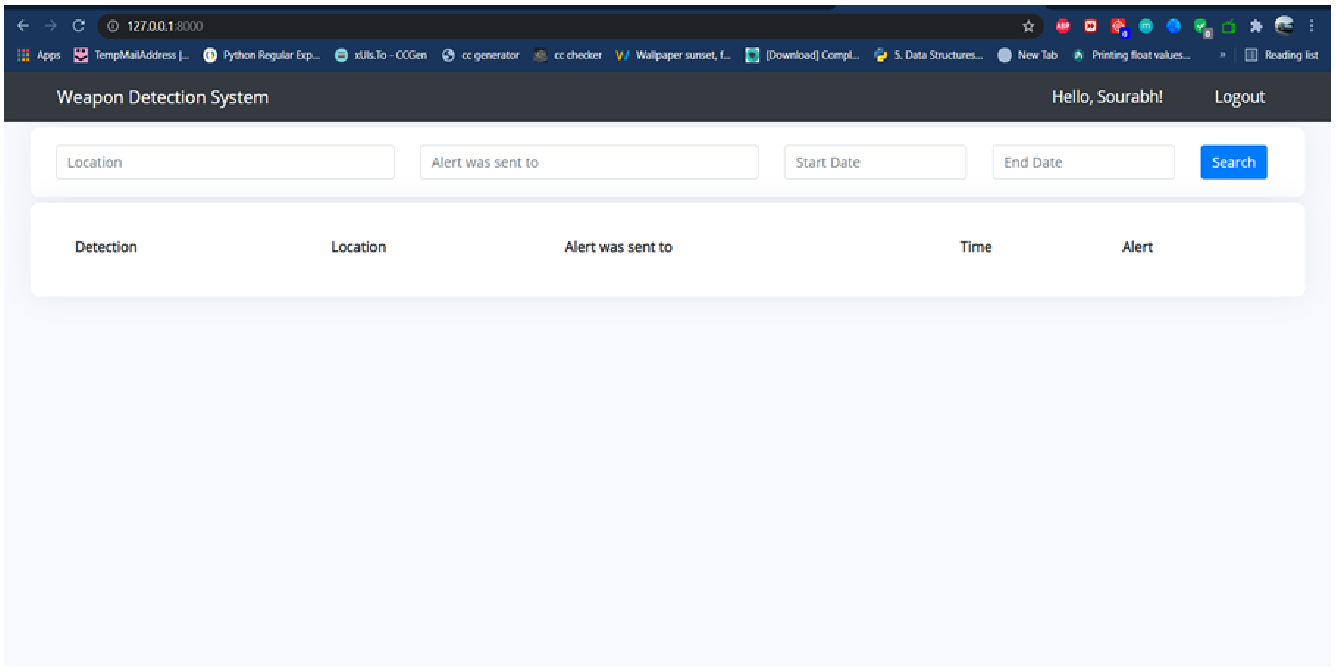
SIGN IN

Account was successfully created for Sourabh

Forgot Password?

Create an account? Sign up

➢ **Landing on Dashboard Page after Login**



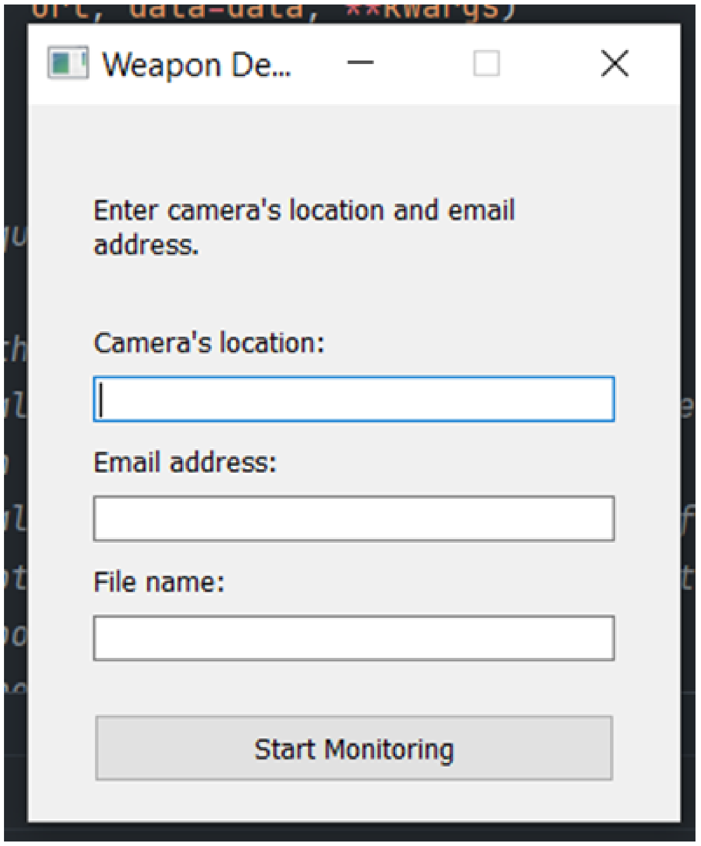
Weapon Detection System

Hello, Sourabh!

Logout

Detection	Location	Alert was sent to	Time	Alert

7.2 CLIENT APPLICATION


Weapon De...

Enter camera's location and email address.

Camera's location:

Email address:

File name:

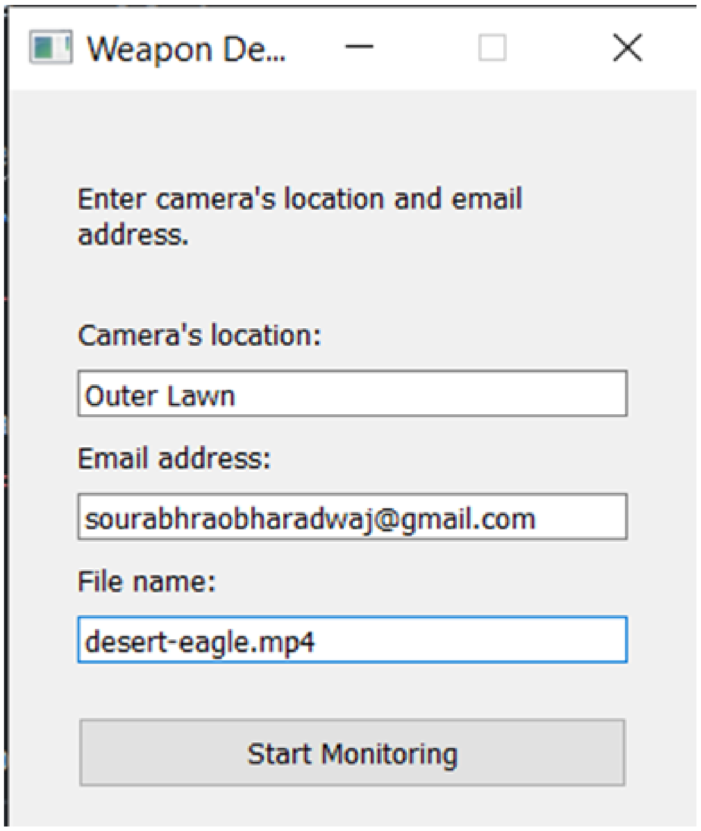
- **Step: 1 Run the Client PyQT5 script**

i). Multiple Platform Support

ii). Just like an App

- **Step: 2 Login to the Application**

Login to the application using registered Email and Password .If you don't have one then please signup for an account in web app.


Weapon De...

Enter camera's location and email address.

Camera's location: Outer Lawn

Email address: sourabhraobharadwaj@gmail.com

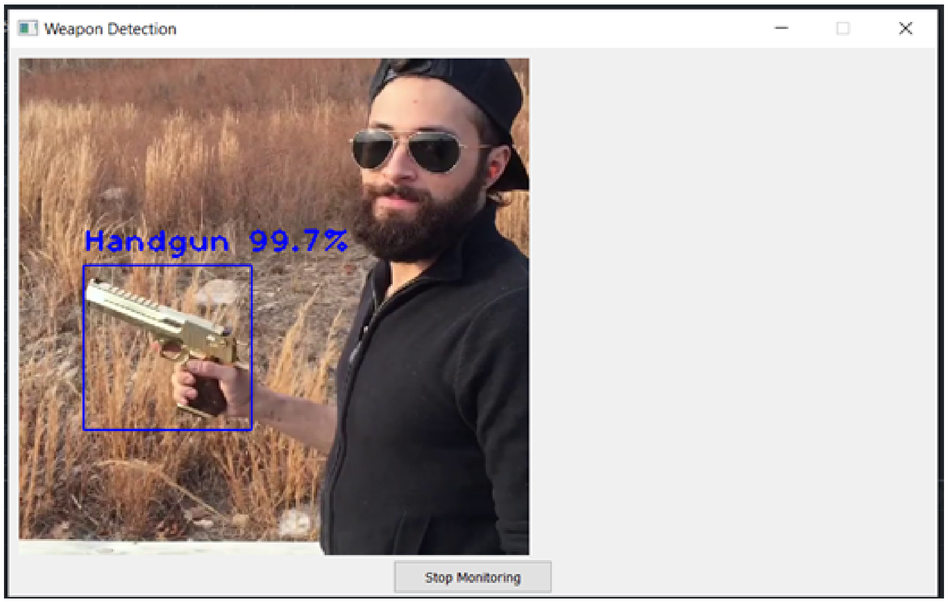
File name: desert-eagle.mp4

- **Step: 3 Provide Information of Data Feed**

Provide camera's location, email id to be notified and video source

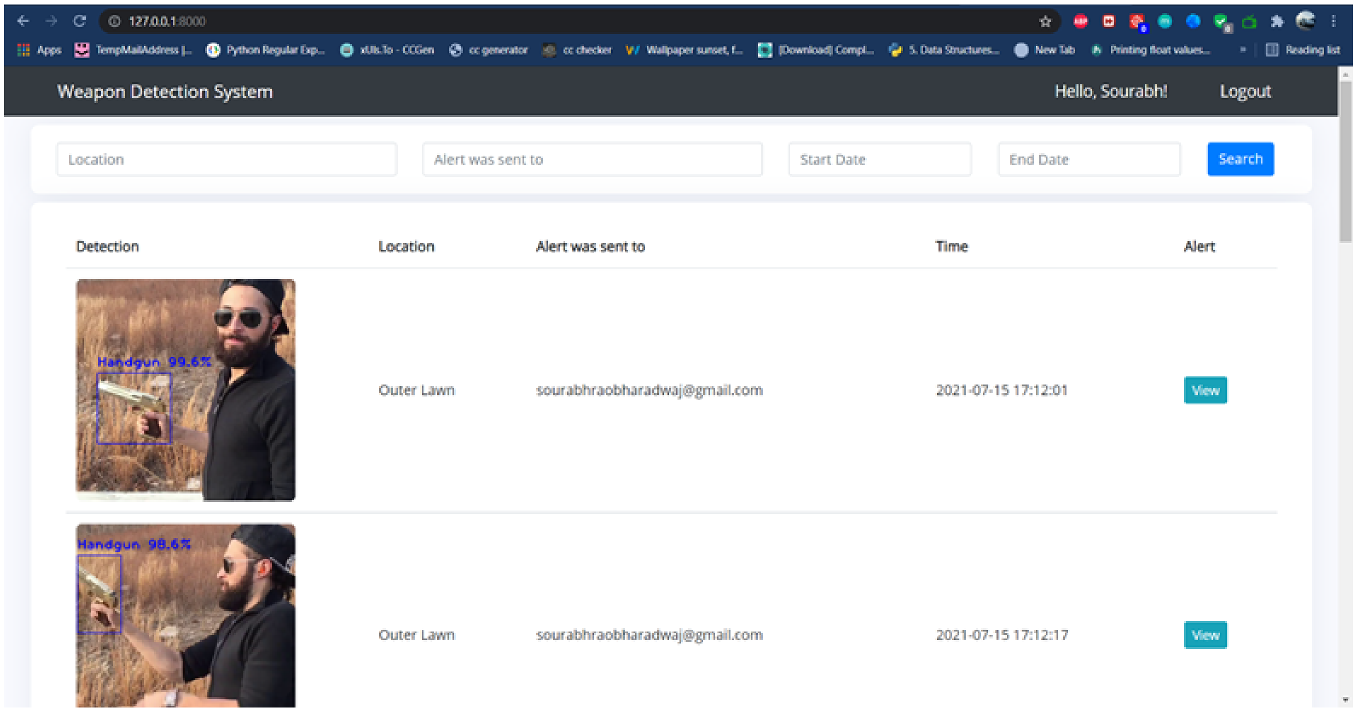
- **Step: 4 Surveillance starts**

Monitoring is Started and if Any weapon is detected within the frame,it will be uploaded to DB and an email will be sent to user



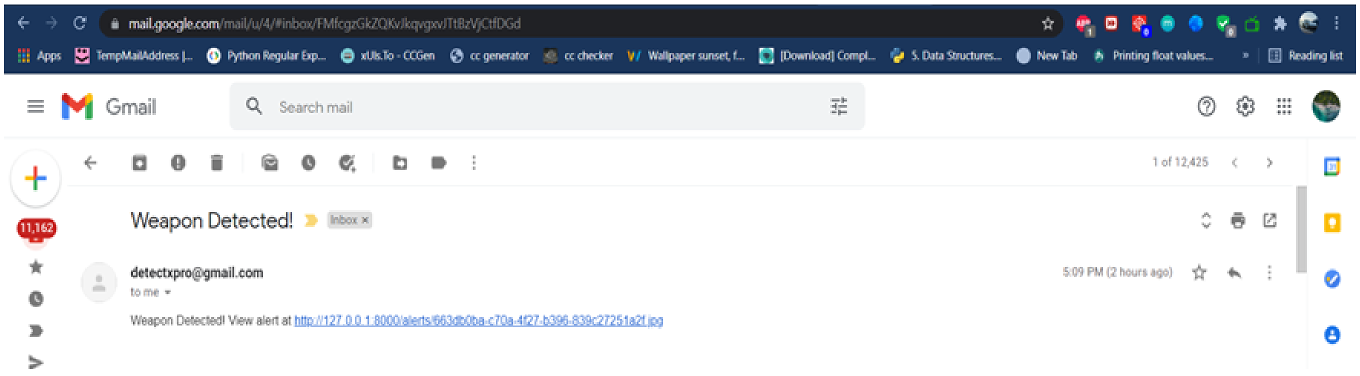
➢ Step: 5 Openback webapp

Login back to web app and view the frame at which weapon was detected



Detection	Location	Alert was sent to	Time	Alert
Handgun 99.6%	Outer Lawn	sourabhraobharadwaj@gmail.com	2021-07-15 17:12:01	View
Handgun 98.6%	Outer Lawn	sourabhraobharadwaj@gmail.com	2021-07-15 17:12:17	View

➢ **An Email will be sent-over registered email to notify**



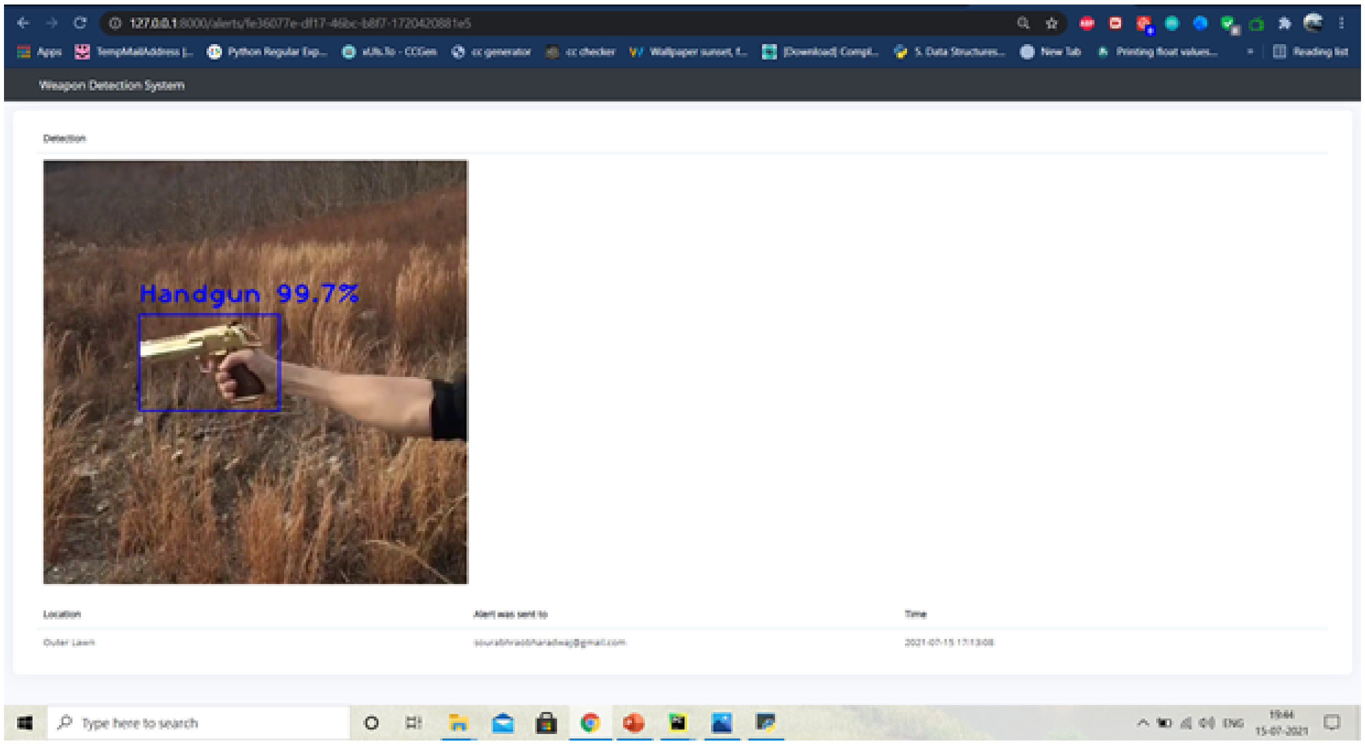
Weapon Detected!

detectxpro@gmail.com

to me

5:09 PM (2 hours ago)

Weapon Detected! View alert at http://127.0.0.1:8000/alerts/663db0ba-c70a-4f27-b396-839c27251a2f.jpg



Weapon Detection System

Detection

Location	Alert was sent to	Time
Outer Lawn	sourabhradbharadwaj@gmail.com	2021-07-15 17:13:08

CONCLUSION AND FUTURE ENHANCEMENT

A real-time frame-based efficient Weapon detection deep learning model has been presented with a high accuracy metric.The Darknet53 model might be bulky but has a good detection capability. The detections per frame are appropriate for real-time monitoring and can be deployed on any GPU based system.
This tool can be used by Security officials to perform Surveillance ,by feeding CCTV footage directly in real time . If current tool is integrated by Military/Security Department, an alert system for nearby police station can be implemented.Although, we presume that by implementing the novel method, the performance of our system, can be refined and its real time processing essentials like complexity of space and time can be diminished.

The Future enhancements are:

- Real-Time implementation with Mask-RCNN.
- Unified loss metric(like Mean Average Precision) for all the three approaches
- Detecting categories of weapons and identifying fake from real ones
- Detecting handheld guns using Pose and Human Key point Estimation.
- Ensemble techniques for detection combining Instance segmentation with Human Keypoint estimation
- Creating a model API and pushing it to realtime deployment in a drone or CCTV using a Raspberry PI
- Trying out more faster algorithms like MXNet or MobileNet for gun detection.
- Trying out RNN based approach for creating automated annotations(DEXTR and PolygonRNN++)

BIBILIOGRAPHY

- Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC 2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
- Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR abs/1311.2524 (2013). http://arxiv.org/abs/1311.2524
- Goutte, C., Gaussier, E.: A probabilistic interpretation of precision, recall and F-Score, with implication for evaluation. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 345–359. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31865-1_25
- Gutoski, M., Ribeiro, M., Aquino, N.M.R., Lazzaretti, A.E., Lopes, H.S.: A clustering-based deep autoencoder for one-class image classification. In: 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), pp. 1–6 (2017)
- Hofer-Schmitz, K., Nguyen, P.H., Berwanger, K.: One-class Autoencoder approach to classify Raman spectra outliers. In: European Symposium on Artificial Neural Networks, ESANN 2018, pp. 189–194 (2018)
- Kotevski, Z., Mitrevski, P.: Experimental comparison of PSNR and SSIM metrics for video quality estimation. In: Davcev, D., Gómez, J.M. (eds.) ICT Innovations 2009, pp. 357–366. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-10781-8_37
- Liu, Wet al.: SSD: single shot multibox detector. CoRR abs/1512.02325 (2015). http://arxiv.org/abs/1512.02325
- Olmos, R., Tabik, S., Herrera, F.: Automatic handgun detection alarm in videos using deep learning. CoRR abs/1702.05147 (2017). http://arxiv.org/abs/1702.05147
- Raghunandan, A., Mohana, M., Pakala, R., Aradhya, H.V.R.: Object detection algorithms for video surveillance applications. In: IEEE - 7th International Conference on Communication and Signal Processing, April 2018. https://doi.org/10.1109/ICCSP.2018.8524461